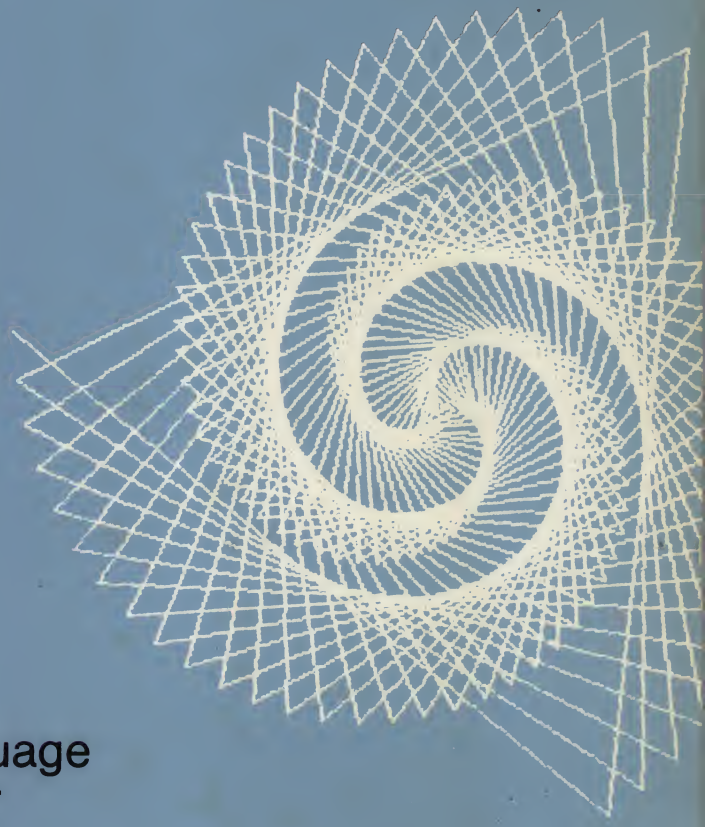
ATARI®

ATARI Logo™ Sourcebook



A Programming Language
for the ST™ Computer

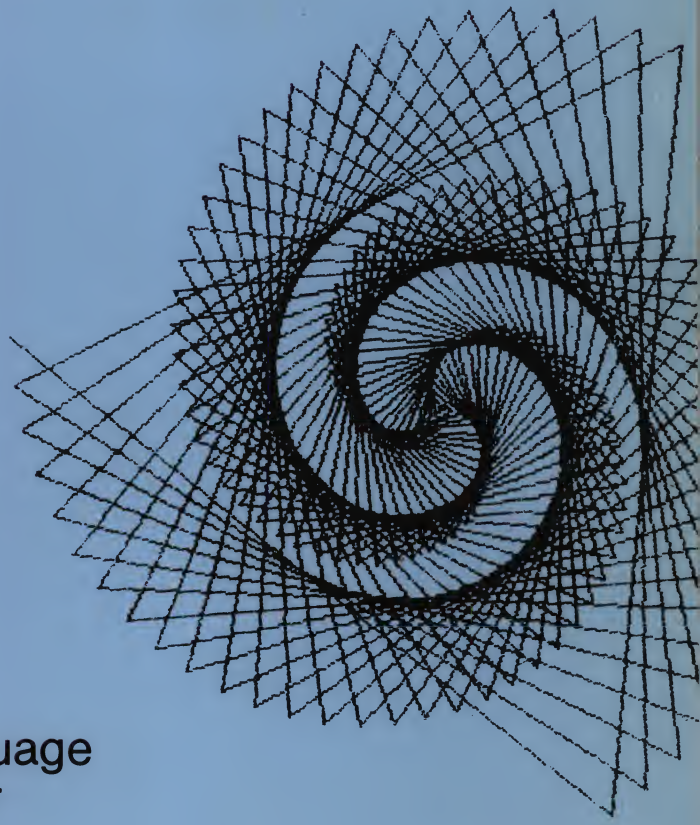
Every effort has been made to ensure the accuracy of the product documentation in this manual. However, because we are constantly improving and updating our computer software and hardware, Atari Corp. is unable to guarantee the accuracy of printed material after the date of publication and disclaims liability for changes, errors, and omissions.

ATARI, ATARI Logo, SC1224, ST, TOS, 260ST, 520ST and 520ST$^+$ are trademarks or registered trademarks of Atari Corp.

GEM is a trademark of Digital Research Inc.

ATARI®

ATARI Logo™ Sourcebook



A Programming Language
for the ST™ Computer

INTRODUCING ATARI LOGO

Logo is a high-level programming language, best known for its use as a teaching tool with beginning computer programmers. ATARI Logo is an enhanced version of the popular language, created to take advantage of the special characteristics of the ATARI ST Computer System and the GEM™ Desktop.

This manual is designed as a sourcebook for ATARI Logo, showing how to use the language in the unique environment of the GEM Desktop. It is recommended that you use this manual as a companion to the *ATARI ST Owner's Manual*. However, this Logo manual is not a general introduction to the language. The first-time programmer should refer to an introductory book or a tutorial on programming with Logo.

This manual is arranged for easy access to all the reference information you need to start programming with ATARI Logo. Chapter 1, *Getting Started with ATARI Logo*, shows you how to make a Backup copy of the ST Language disk and how to load your Logo program into the ST Computer. Chapter 2, *ATARI Logo and GEM*, is a general introduction to Logo. Chapter 3, *ATARI Logo Menus*, provides a detailed explanation of each Logo menu option. And the Appendices contain all the reference materials the programmer will need, from a complete list of Logo primitives to an Error Message listing. Also, Appendix I shows sample procedures for each of the unique primitives that were developed specifically for ATARI Logo.

TABLE OF CONTENTS

CHAPTER 1
GETTING STARTED WITH ATARI LOGO

Making A Backup Disk

Before you begin programming with ATARI Logo on your ST Computer, you should make a backup copy of the program. Having a Backup disk provides security against accidentally erasing or damaging your program language disk.

To make a Backup disk you will need a new, blank, 3-1/2 inch disk. (Disks can be purchased at any computer retailer.) Making a Backup disk is very easy. With the computer turned on and the TOS™ System disk loaded, just follow the steps below and read the prompts that appear in the Dialog Boxes.

1. If you have one disk drive, remove the TOS System disk and insert the Backup disk into the disk drive. When the ST Computer requires a disk switch, it will display the message in a Dialog Box.

Note: If you have two disk drives, insert the ST Language disk into Drive A and the Backup disk into Drive B. With two disk drives, always keep the ST Language disk in Drive A and the Backup disk in Drive B, and follow the prompts.

2. To format the Backup disk, click on the icon for Floppy Disk B and select the Format option from the File heading in the Menu Bar. Click on the left mouse button and the first Format Box will be displayed. A message warning that formatting the disk erases the information on the disk is provided. (The messages will refer to the Backup disk as Disk B and the ST Language disk as Disk A.) Click on the OK button and proceed to the second Format Box.

You can label the disk with the second Format Box. Type in a descriptive name, like "Logo". Make sure the option "single-sided" is shaded, then click on the Format button. You already have your Backup disk in Drive A, so continue to Step 3.

As the disk is being formatted, you will be able to watch the process in the Working Box. When the disk is formatted, a Dialog Box stating that the disk can how hold 357,376 bytes of information will be displayed. Click on the Exit button.

You will be returned to the Format Box. Click on the Exit button and return to the GEM Desktop.

Note: If the formatted disk cannot hold 357,376 bytes of information, it is a defective disk. Place a new disk in the disk drive and format it.

3. To copy the ST Language disk, insert it into your disk drive, select Floppy Disk A with the mouse pointer and drag it on top of the Floppy Disk B icon, and release the left mouse button.

A Dialog Box warning that copying Disk A to Disk B erases all information on Disk B will be displayed. Click the OK button and the Diskcopy Box will be displayed.

Click on the Copy button and follow the prompts to finish the copying process. You will be instructed to switch disks until the copying process is completed.

If you have any questions or problems making a Backup copy of your ST Language disk, refer to the *ATARI ST Owner's Manual* for detailed information.

Loading ATARI Logo

To begin using ATARI Logo, you need to load the Logo program into your ST Computer. Follow the instructions below to load ATARI Logo into your computer for either a one- or two-drive computer system.

With One Disk Drive

1. With the ST Computer turned on and the GEM Desktop on the video display screen, double-click on the Floppy Disk B icon.

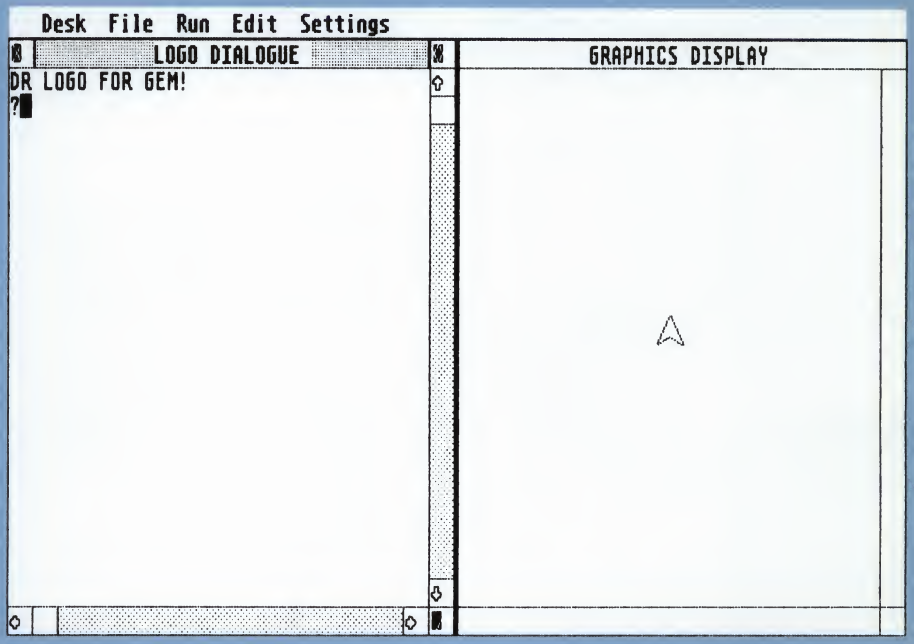
2. When the Dialog Box requests that you insert Disk B into Drive A, place the ST Language disk into Drive A and press the **[Return]** key.

3. When the Floppy Disk window opens, double-click on the LOGO.PRG icon and the Logo Desktop will appear on the video display screen.

With Two Disk Drives

1. With the ST Computer turned on and the GEM Desktop on the video display screen, insert the ST Language disk into Drive B and double-click on the Floppy Disk B icon.

2. When the Floppy Disk B window opens, double-click on the LOGO.PRG icon and the Logo Desktop will appear on the video display screen.



The Logo Desktop is the main point of reference for all your work with ATARI Logo. Look over Chapter 2, *ATARI Logo and GEM*, for information on the Logo Desktop and the Logo Windows.

CHAPTER 2
ATARI LOGO AND GEM

ATARI Logo uses the standard operating procedures of the GEM Desktop. The procedures for accessing menu items, selecting options, manipulating windows, and loading applications are all explained in detail in the *ATARI ST Owner's Manual*.

Menus

The ATARI Logo menus are accessible from the Menu Bar which borders the top edge of the Logo Desktop. The menu options available under the Desk heading are identical to those available from the GEM Desktop. The other menu headings are specific to ATARI Logo and are explained in Chapter 3, *ATARI Logo Menus*.

Dialog Boxes and Error Messages

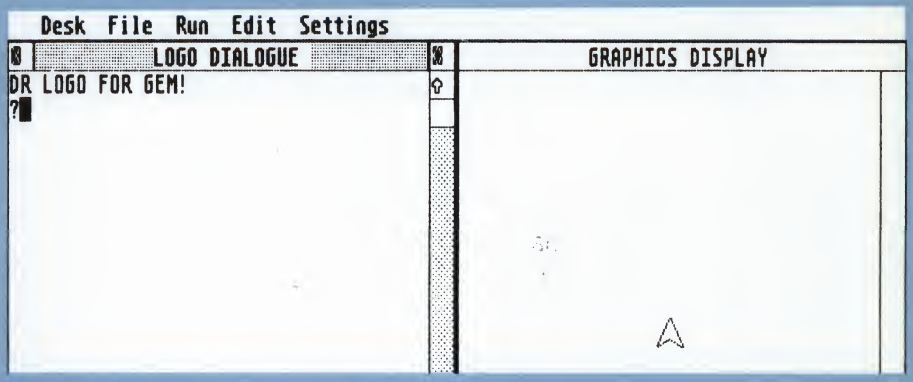
Dialog Boxes will appear in the center of the Logo Desktop whenever the program requires information that is not being provided in the program listing. Error Messages are also presented in a Dialog Box. Whenever an Error Message appears, information concerning a Logo format or procedure will be provided. For a complete listing of ATARI Logo Error Messages, refer to Appendix E.

To exit from a Dialog Box, point at one of the Exit buttons and click the left mouse button. If the Exit button has an enlarged border, you can press the **[Return]** key on the ST keyboard rather than using the left mouse button.

Windows

The procedures for sizing, moving, opening, closing, scrolling, and managing multiple windows are identical to the methods described in Chapter 4 of the *ATARI ST Owner's Manual*. Please refer to that manual for specific information.

The Logo Desktop is divided into two windows: The Logo Dialogue Window and the Graphics Display Window.



When you write or load a Logo program, the program listing will appear in the Logo Dialogue Window. The corresponding picture will appear in the Graphics Display Window.

The other two windows available with ATARI Logo are the Edit Window and the Debug Window. The Edit Window will open whenever you enter an edit command. (Refer to Appendix G for a list of Logo editing procedures and variables.) The Edit Window is the workspace for editing your program procedures. Edit changes made within the Edit Window can then be used and stored on a floppy disk.

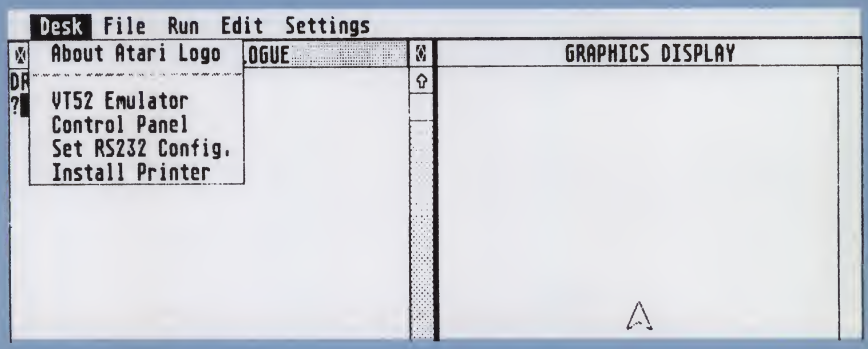
The Debug Window allows you to see a program listing as the program is running. For more information on the Debug Window, refer to Chapter 3, *ATARI Logo Menus,* under the menu items Trace and Watch.

CHAPTER 3
ATARI LOGO MENUS

Along the top edge of the Logo Desktop is the Menu Bar. The Menu headings are Desk, File, Run, Edit, and Settings. Each heading has its own menu. To access a menu from the Menu Bar, point at the Menu heading. The word will become shaded and the menu will automatically drop down. If you don't want to select a menu item, click anywhere else on the Logo Desktop and the menu will pop back up.

Desk

The Desk menu contains options that are available from ATARI Logo and from within most application programs that run on the ST Computer.



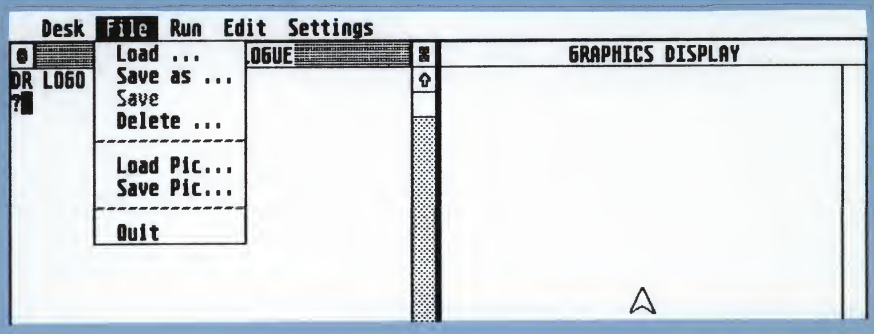
About ATARI Logo

This option is the billboard for the application program. Copyright and general program information are displayed. Select the About ATARI Logo option and the following Dialog Box will be displayed:

The other options in the Desk menu—VT52 Emulator, Control Panel, Set RS232 Configuration, and Install Printer—are explained in detail in the *ATARI ST Owner's Manual*. Refer to the section on each option in Chapter 5, *The Opening Menu*, of that manual.

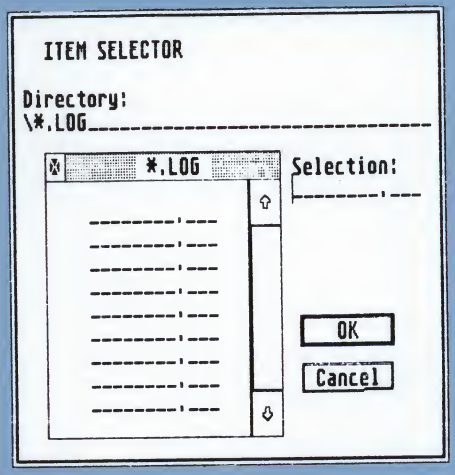
File

The File menu contains options that let you read information from and write information to the disk drive.

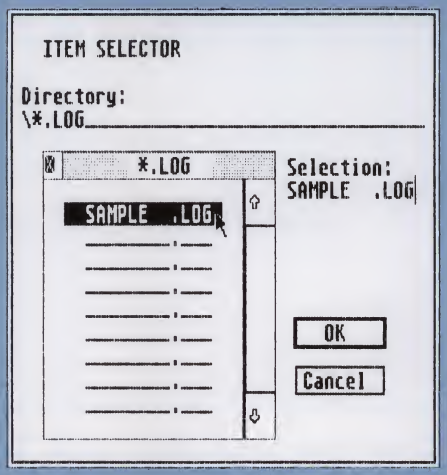


Load

The Load option reads a file that you have stored on a floppy disk. Select the Load option and the following Dialog Box will be displayed:



To select a file listed in the Item Selector Box, point at a filename and double-click the left mouse button. You can also select a file by clicking once on an item and then clicking once on the OK button.



The current directory is displayed at the top of the Item Selector Box under the heading "Directory". If the file you want to access is stored in a different directory you can change the directory. Click on the Directory heading, use the **[Backspace]** key to erase the current directory name, and type in the name of the directory you want to use. To view a listing of the files under the new directory, simply click anywhere inside the Directory Window and the new directory listing will appear.

If you decide not to load a particular file, or if the file you want is not present, you can exit the Item Selector Box by clicking on the Cancel button.

Save As

The Save As option creates a new file. You can also use this option to make a copy of a file using a different name. Each time you select the Save As option, the Item Selector Box will be displayed.

To enter the filename of the new file in the Item Selector Box, type it in on the ST keyboard. To delete characters, use the **[Backspace]** key.

Note: You cannot use an existing filename when you name or rename a file.

Save

The Save option writes the contents of the Logo Desktop onto a floppy disk. This option can only be used if you have previously stored the file you are working on. If you are working with a new file, the word "Save" on the File menu will be displayed in lightened letters. The lightened letters indicate that the option is not available to you as a new file. To save a new file, use the Save As option.

Delete

The Delete option removes a file from a floppy disk. When you select the Delete option, the Item Selector Box will be displayed. To delete a file, either double-click on the filename, or click on the filename once and then click on the OK button.

Load Pic

The Load Pic option loads the graphic design stored in your file into the Graphics Display Window. When you select the Load Pic option, the Item Selector Box will be displayed. To load a file, select the filename by either double-clicking the left mouse button or clicking once on the left mouse button and then clicking on the OK button.

The Graphics Display Window will automatically size itself to the dimensions of the picture that is being loaded.

Note: The saved file must be in the same resolution as you are currently working in or an Error Message will be displayed.

Save Pic

The Save Pic option lets you store pictures that you have created with Logo. The picture that is currently in the Graphics Display Window is placed into a file on a floppy disk.

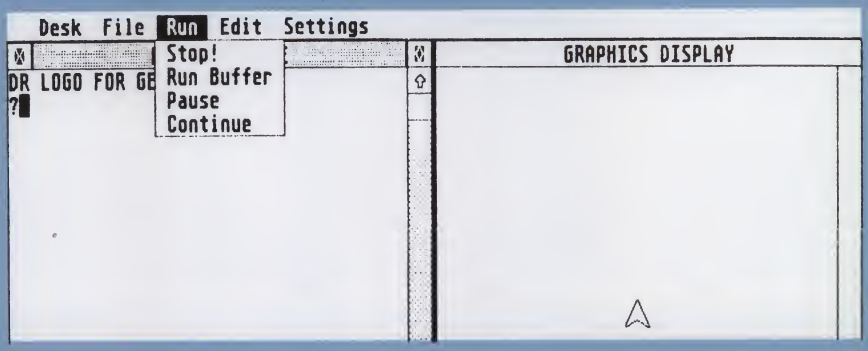
When you select the Save Pic option, the Item Selector Box will be displayed. Type in the name of the file you wish to save, and click on the OK button.

Quit

The Quit option lets you exit ATARI Logo. Before returning to the Logo Desktop, you will be asked if you have saved all of your work. If you have not, you may click the Cancel button and then save any files you wish. If you don't want to save any of your work, click on the OK button and you will be returned to the Desktop.

Run

The Run menu provides options that are used to control the starting and stopping of the procedures you use with ATARI Logo.



Stop

The Stop option will terminate whatever procedure you are running or editing, and return you to the Logo Desktop. The Stop option has the same effect as pressing **[Control] [G]** on the ST Computer keyboard.

Run Buffer

Each time you enter a command or set of commands into the ST Computer, Logo stores that command line in a memory buffer. When you select the Run Buffer option, Logo runs that command again. For example, if you type the command:

FD 100 [Return]

the turtle will move forward the specified distance. Selecting the Run Buffer option will make the turtle move that distance forward again.

Pause

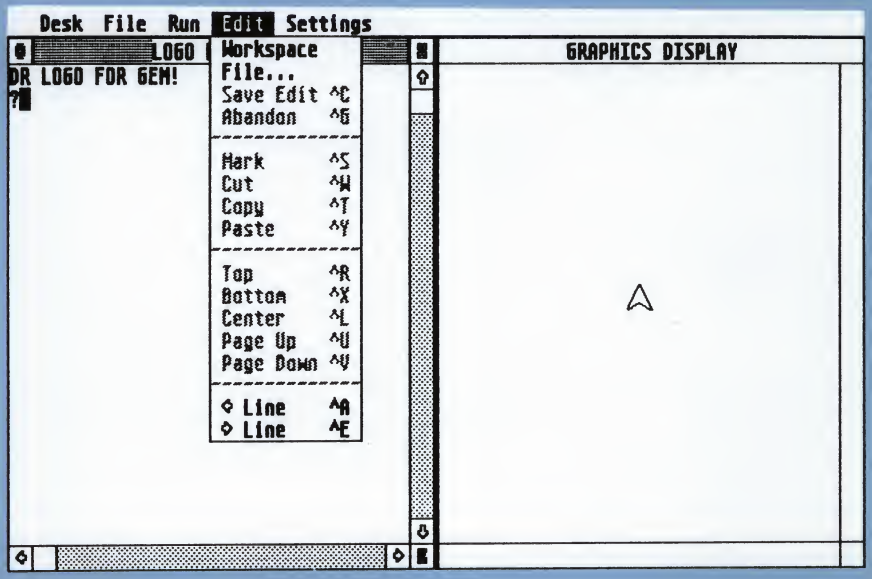
The Pause option temporarily halts a procedure that is currently running. Using the Pause option allows you to enter information with the keyboard onto the Logo Desktop. After you have entered the information you want, you can return to your procedure by entering the keyboard command CONTINUE, or its abbreviation, CO.

Continue

When you select the Continue option, a procedure that was temporarily halted with a PAUSE command will continue to run. This option serves the same purpose as the keyboard command CONTINUE(CO).

Edit

The Edit menu controls all of the editing capabilities of ATARI Logo.



Workspace

The Workspace option places all of the procedures that you have entered into the ST Computer into the Edit Window so you can edit them.

File

The File option lets you load a file from a floppy disk and place it into the Edit Window. When you select the File option, the Item Selector Box will be displayed and you can choose the file to be loaded. After being edited, this file can be discarded by pressing **[Control] [G]** or saved to disk by pressing **[Control] [C]**.

Save Edit ^C

The Save Edit option transfers the material you are editing onto the Logo Desktop.

The Save Edit option can also be accessed by pressing [Control] [C] on the St keyboard.

Abandon ^G

The Abandon option exits from running a program or from editing a procedure, and then returns to the Dialogue Window.

Warning: If you use the Abandon option from the Edit Window you will lose whatever material you were editing.

The Abandon option can also be accessed by pressing [Control] [G] on the ST keyboard.

Mark ^S

While you are editing a procedure, you can cut (remove), copy, or paste (insert) parts of the procedure. You need to mark the section of a procedure you want to manipulate. Use the Mark option to mark the beginning and end of a section.

To mark a section of a procedure, move the cursor so it is positioned just before the section you want to mark and select the Mark option. Then move the cursor to the end of the section you want to mark and select the Mark option again.

The Mark option can also be accessed by pressing [Control] [S] on the St keyboard.

Cut ^W

The Cut option deletes a section of a procedure after the section is marked. If you delete a section and change your mind, you can put that section back (if you haven't selected another section) by selecting the Paste option.

The Cut option can also be accessed by pressing [Control] [W] on the ST keyboard.

Copy ^T

The Copy option places a copy of the marked section into the procedure.

The Copy option can also be accessed by pressing [Control] [T] on the ST keyboard.

Paste ^Y

The Paste option places the cut section into the procedure.

The Paste option can also be accessed by pressing [Control] [Y] on the ST keyboard.

Top ^R

The Top option moves the cursor to the top of the procedure in the Edit Window.

The Top option can also be accessed by pressing [Control] [R] on the ST keyboard.

Bottom ^X

The Bottom option moves the cursor to the bottom of the procedure in the Edit Window.

The Bottom option can also be accessed by pressing [Control] [X] on the ST keyboard.

Center ^L

The Center option scrolls the line indicated by the cursor to the center of the Edit Window.

The Center option can also be accessed by pressing [Control] [L] on the ST keyboard.

Page Up ^U

The Page Up option scrolls the procedure in the Edit Window up one window full of text.

The Page Up option can also be accessed by pressing [Control] [U] on the ST keyboard.

Page Down ^V

The Page Down option scrolls the procedure in the Edit Window down one window full of text.

The Page Down option can also be accessed by pressing [Control] [V] on the ST keyboard.

< = Line ^A

The < = Line option moves the cursor to the beginning of the line.

The < = Line option can also be accessed by pressing [Control] [A] on the ST keyboard.

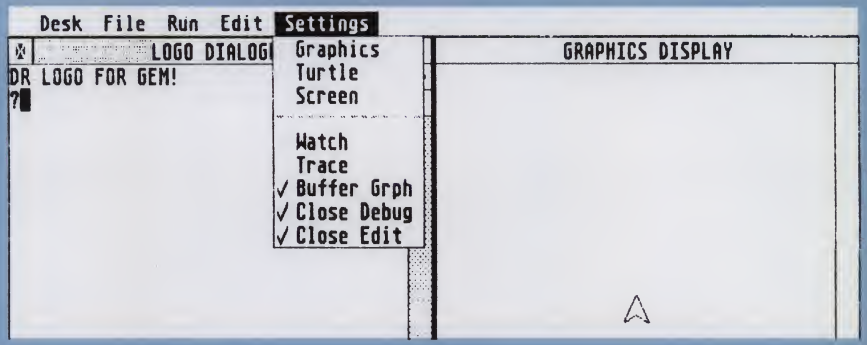
= > Line ^E

The = > Line option moves the cursor to the end of the line.

The = > Line option can also be accessed by pressing [Control] [E] on the ST keyboard.

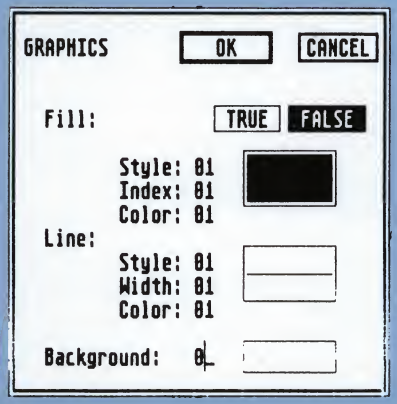
Settings

The Settings menu options control many of the major operating options of ATARI Logo. With these menu options you can change the line and fill patterns, determine which windows open and close, and define the movements of the turtle.



Graphics

Select the Graphics option and the Graphics Dialog Box will be displayed.



The three graphics functions that you can customize with this menu option are: Fill style and color (if you have a color monitor); Line style and color; and the Background color.

You can also determine whether the ATARI Logo primitives will be drawn filled with the new fill style and color, or hollow with the current line attributes. Select the TRUE button and the graphics functions you define in the Graphics Dialog Box will apply to your primitives. Select the FALSE button and the Logo primitives will be drawn without being filled.

The Graphics Dialog Box lets you define the graphics functions and preview what the functions will look like. As you set the graphics field, the corresponding representations can be seen by clicking inside of the Preview Windows. The Preview Windows are directly to the right of the settings options.

To set a field, point and click at the setting you want to change. The cursor (a vertical line) will appear and you can type in the number of the setting. The new setting can be viewed by clicking inside of the Preview Window. To change the setting, type in a different number. To change a different setting option, point at the setting and click the mouse button. The cursor will move to the new setting.

To integrate your new graphics functions into the Logo program, select the OK button. To cancel your new settings, select the CANCEL button.

Fill The Fill characteristics can be defined by Style, Index, and Color.

There are five settings for Style:

0 = Hollow
1 = Solid
2 = Pattern
3 = Hatch
4 = User Defined

The Index setting is used only when you choose either Style Setting 2 (Pattern) or 3 (Hatch). With the combination of Style 2 and Index, you have 24 available patterns. Using Style 3 and Index, you have 12 different Hatch patterns available. Choose Style 2 or 3, then try different Index numbers to see the possible variations.

When you choose Style Setting 4 (User Defined), you can create a custom pattern. To create a custom pattern, enter the following line into the Logo Dialogue Window:

PPROP "GRAPHICS ".FPT [n1 n2 n3 . . . n16]

The numbers n1 through n16 determine the actual pattern. These numbers can be any whole number between 0 and 65535. The pattern will be made up of the binary representation of the numbers used.

As an example, enter the following version of the custom pattern. To modify the pattern, experiment by changing the numbers.

PPROP "GRAPHICS ".FPT [0 0 128 448 992 2032 4088 8188 16382 8188 4088 2302 992 448 1280]

The Color setting is either 1 (black) or 0 (white) with a monochrome monitor. With an ATARI RGB Color Monitor, you can choose four colors (0-4) or 16 colors (0-15) depending on the resolution you are working with. To adjust the colors, you can use the Logo SETPAL command or the Control Panel. Refer to the section on the Control Panel in Chapter 5 of the *ATARI ST Owner's Manual* for more information.

Line The Line characteristics can be defined by Style, Width, and Color.

The Style Setting has seven different styles available (1-7). When you choose Style 1, you have 39 available line widths. Choose Style 1, then select a Width from 1-39.

Line Style 7 is the user defined setting. To create a custom line pattern, enter the following line into the Logo Dialogue Box:

PPROP "GRAPHICS ".LPT n1

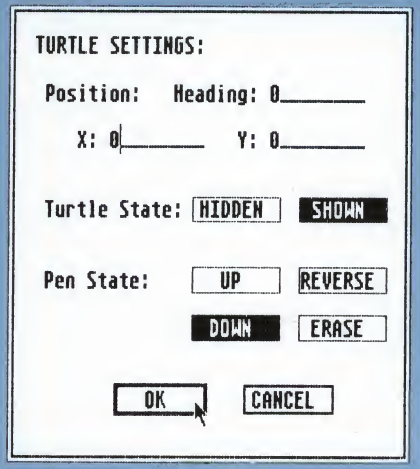
Experiment by changing the number to create different patterns.

The Color setting for Line is set exactly the same as for Fill.

Background The Background setting establishes the color of the background. The Color setting is either 1 (black) or 0 (white) with a monochrome monitor. With the ATARI SC1224™ RGB Color Monitor, you can choose four colors (0-3) or 16 colors (0-15) depending on the resolution you are working with. To adjust the colors, refer to the section on the Control Panel in Chapter 5 of the *ATARI ST Owner's Manual*.

Turtle

When you select the Turtle option from the Settings menu, the Turtle Settings Box is displayed.



The Turtle Settings Box allows you to control the Position and State of the turtle, and the Pen State. You can also refer to this Dialog Box to view the current status of the turtle.

To set a function in the Turtle Settings Box, point and click at the setting you want to change. The cursor (a vertical line) will appear and you can type in the number of the setting.

Turtle Heading To change the turtle's Heading (direction), enter a number into the space following "Heading". The number 0 will instruct the turtle to head straight upwards. Other numbers will indicate different directions.

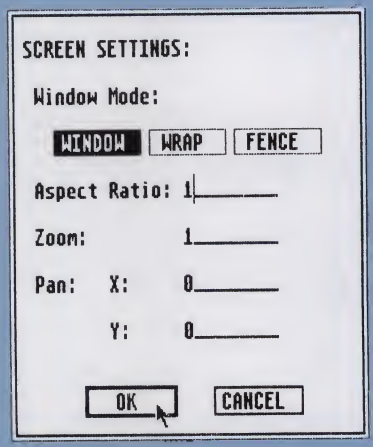
Turtle Position To set the turtle's Position, enter numbers in the X and Y coordinate positions. The coordinates 0,0 are at the center of the Graphics Display Window.

Turtle State and Pen State To set the Turtle State and Pen State, select the setting you want by clicking on the button for the setting.

When you have established all the settings, select either the OK button to confirm your selections, or the CANCEL button to disregard them.

Screen

Select the Screen option from the Settings menu and the Screen Settings Box will be displayed.



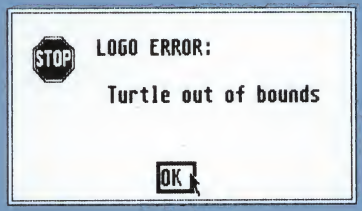
The Screen Settings Box allows you to control the Window Mode, the Aspect Ratio, and the Zoom and Pan values.

To set a function in the Screen Settings Box, point and click at the setting you want to change. The cursor (a vertical line) will appear and you can type in the number of the setting.

Window Mode There are three settings for the Window Mode: WINDOW, WRAP, and FENCE. To select one of the settings, simply click on the appropriate button.

The WINDOW setting lets the turtle draw beyond the edge of the window.

The WRAP and FENCE settings constrain the turtle to the screen in different ways. The WRAP setting will allow the turtle to draw off the edge of the screen area by making it reappear on the opposite side of the screen. When you use the FENCE setting, entering a command that sends the turtle off the edge of the screen gives you the following Error Message:



Aspect Ratio Changing the Aspect Ratio affects the shape of the objects you draw. For example, if you choose an Aspect Ratio of 1 and draw an ellipse in the middle of the screen, and then change the Aspect Ratio to .5, the ellipse will be flattened. Changing the Aspect Ratio to 1.5 will change the shape of the ellipse in the opposite direction.

Zoom The Zoom function changes the size of the pictures drawn by scaling down the values of x and y. For example, the standard Zoom value is 1. If you draw a circle in standard Zoom mode and change the Zoom setting to 2, the next time you draw a circle, the proportions will be twice as large. Changing the Zoom setting to .5 will change the proportions by one half.

Pan The Pan function changes the location of the picture in relation to the center of the screen. Setting the Pan coordinates establishes the starting point for any picture you draw within the Graphics Display Window.

Watch

The Watch option on the Settings menu opens the Debug Window. Within the Debug Window, the steps of your procedure are displayed as they are executed.

Trace

The Trace option on the Settings menu also opens the Debug Window. You can determine what values are being assigned to your variables at any time while your program is running.

Buffer Grph

Whenever you draw something in the Graphics Display Window, that image is saved in a buffer. If you open another window over the Graphics Window and then close the window, the image in the Graphics Window will be redrawn.

If Logo cannot find enough memory for a buffer, it will send the graphics to your floppy disk. Sending the graphics to the floppy disk is a time consuming operation, so you may consider turning off the buffer if this happens. You can turn off the redrawing procedure by selecting Buffer Grph. The Buffer Grph is off when there isn't a check mark in front of the option.

Close Debug

The Close Debug option lets the Debug Window remain open after the debugging procedure is completed. The Close Debug option is selected when a check mark is visible in front of the option on the menu.

Close Edit

The Close Edit option lets the Edit Window remain open after the editing procedure is completed. The Close Edit option is selected when a check mark is visible in front of the option on the menu.

APPENDIX A
LOGO CONTROL CHARACTER COMMANDS

Use the Control Character Commands to control the screen display and cursor movements. To enter a Control Character Command, hold down the Control key and press the indicated letter key.

Character	Effect
Ctrl-A	Moves the cursor to the beginning of the line.
Ctrl-B	Moves the cursor one position to the left.
Ctrl-C *	Exits the text editor and updates the Logo workspace with the definitions of all the procedures and variables in the text editor's buffer.
Ctrl-E	Moves the cursor to the end of the line.
Ctrl-F	Moves the cursor one position to the right.
Ctrl-G	When outside the text editor, **[Control] [G]** immediately terminates the current procedure. When inside the text editor, it exits the text editor without updating the Logo workspace and discards any changes made during the text editing session.
Ctrl-H	Deletes the character to the left of the cursor.
Ctrl-I	Moves the cursor to the next tab setting (column 5, 9, 13 . . .) and inserts up to 4 spaces in the current line.
Ctrl-K	Deletes all characters to the right of the cursor. Deleted characters are stored in a buffer and can be restored with a **[Control] [Y]**.

*Indicates the character is valid only within the text editor.

Ctrl-L *	When inside the text editor, **[Control] [L]** readjusts the display so that the line currently indicated by the cursor is positioned at the center of the window. If the cursor is less than 12 lines from the beginning of the buffer, the text editor redisplays the window when **[Control] [L]** is pressed.
Ctrl-M	Generates a carriage return and enters information into the computer.
Ctrl-N	Moves the cursor to the next line in the text editor. The cursor moves down one line towards the end of the buffer.
Ctrl-O *	Opens a new line in the text editor. It is equivalent to pressing **[Enter]** followed by **[Control] [B]**.
Ctrl-P	Moves the cursor to the previous line and the cursor moves up one line towards the beginning of the buffer.
Ctrl-Q	Generates the quoting character # that makes Logo treat a delimiter character as a literal character. Delimiter characters are: [ ] ( ) " : ; = < > + / ^.
Ctrl-R *	Positions the cursor at the beginning of the text editor's buffer.
Ctrl-S *	Marks block.
Ctrl-T *	Copies block.
Ctrl-U *	Displays the previous page of text in the text editor's buffer.
Ctrl-V *	Displays the next page of text in the text editor's buffer.
Ctrl-W *	Deletes (cut) block.
Ctrl-X *	Positions the cursor at the end of the text editor's buffer.

*Indicates the character is valid only within the text editor.

Ctrl-Y	Redisplays the line most recently stored in the buffer by an **[Enter]**, or **[Control] [K]**, or cut/copy.
Ctrl-Z	Interrupts the current procedure and displays a pause prompt to allow interactive debugging. Enter CO to continue the execution of the interrupted procedure; enter THROW "TOPLEVEL to exit to the outer most level; enter STOP to exit to the prior level.

APPENDIX B
LOGO SYSTEM PRIMITIVES

An ATARI Logo system primitive is preceded by a period and allows the user to look at memory locations or manipulate lists.

Primitive	Inputs	Definition and Example
.CONTENTS		Displays the contents of the ATARI Logo symbol space. ?.CONTENTS
.DEPOSIT	n1 n2	Puts n2 into the absolute memory location specified by the first input number. ?.DEPOSIT 2051 7
.EXAMINE	n	Displays the contents of the absolute memory location specified by the input number (byte value). ?.EXAMINE 2051 7
.REPLACE	item n varlist object	Replaces the specified item in the list with the object. The list must be the value of a variable. ?MAKE "VARLIST [A B C D E F] ?.REPLACE 4 :VARLIST [1 2 3] ?:VARLIST [A B C [1 2 3] E F]
.REPTAIL	item_n varlist object	Replaces all items following the specified item in the list with the object. The list must be the value of a variable. ?MAKE "VARLIST [A B C D E F] ?.REPTAIL 4 :VARLIST [1 2 3] ?:VARLIST [A B C D 1 2 3]

APPENDIX C
LOGO SYSTEM VARIABLES

Variables	Definition and Example
ERRACT	When TRUE, causes a pause when an error occurs.
FALSE	System value.
GFILL	If TRUE, graphic objects are filled using current fill attributes.
GRAPHICS	Holds property lists defining user fill and line type patterns. PPROP "GRAPHICS ".FPT [<16 integers>] PPROP "GRAPHICS ".LPT <integer>
PD	Value of turtle's pen state meaning PENDOWN.
PE	Value of turtle's pen state meaning PENERASE.
PU	Value of turtle's pen state meaning PENUP.
PX	Value of turtle's pen state meaning PENREVERSE.
REDEFP	When TRUE allows redefinition of primitives.
TOPLEVEL	Interpreter's prompt (?). When ATARI Logo is at TOPLEVEL a question mark displays, there are no procedures on the stack, and the level number is zero. A THROW will exit all pending procedures.
TRUE	System value.

APPENDIX D
LOGO SYSTEM PROPERTIES

Property Name	Property Value
.APV	Associated Property Value. The value of a global variable.
.BUR	When TRUE, package is buried.
.DEF	Definltion of a procedure.
.ENL	End of a procedure line that is broken by a carriage return and spaces.
.FMT	Beginning of a procedure line that is broken by a carriage return and spaces.
.FPT	Identifies user-defined fill pattern.
.LPT	Identifies user-defined line type.
.PAK	Name of package to which this object belongs.
.PKG	When TRUE, the object is a package name.
.PRM	Identifies a primitive.
.REM	Remark or comment.
.SPC	Space.

APPENDIX E
LOGO ERROR MESSAGES

Message

Number too big
No file selected
(symbol) is a primitive
Can't find LABEL (symbol)
Can't (symbol) from the editor
I'm having trouble with the disk
Disk is full
Can't divide by zero
File is not open
File already exists
File not found
Can't find CATCH for (symbol)
I'm out of space
(symbol) is not true nor false
Not enough inputs to (procedure)
Too few items in (list)
Turtle out of bounds
I don't know how to (symbol)
(symbol) has no value
)without(
I don't know what to do with (symbol)
Primitive is not implemented
Disk is write-protected
(procedure) doesn't like (symbol) as input
(procedure) didn't output
The word is too long
I don't have enough buffer space
IF wants [ ]'s around instruction list
(symbol) isn't a parameter
I can't (symbol) while loading
The file is write-protected
I can't find the disk drive
No PAN with FENCE or WRAP
Error messages for picture files

APPENDIX F
LOGO PRIMITIVES

The Logo primitive and its input(s) are listed alphabetically followed by a definition. Primitive names are entered in uppercase characters.

Primitive	Inputs	Definition and Example
ABS	n	Outputs the absolute value of the input number. ?ABS −3 3
AND	expression, expression...	Outputs TRUE if all input expressions are true. Otherwise it outputs FALSE. ?AND (3<4) (7>4) TRUE
ARC	[x y radius begin_angle end_angle]	Makes ATARI Logo draw an arc at the x- and y- coordinates with the input radius, using the input beginning and ending angles.
ARCTAN	n	Outputs the arc tangent (inverse tangent) of the input number. Outputs are in degrees. ?ARCTAN 2 63.434953
ASCII	word	Outputs the ASCII value of the first character in the input word. ?ASCII "GREEN 71 ASCII "G 71

BACK BK	distance__n	Moves the turtle the input number of steps in the opposite direction of its heading. ?BACK 50
BOX	[x y width height]	Makes Logo draw a box at the x- and y- coordinates with the input width and height.
BURY	pkgname | pkgname__list	Hides the specified package(s) from work space management commands: EDALL, EDNS, EDPS, ERALL, ERNS, ERPS, GLIST, POALL, PONS, POPS, POTS, PPS, SAVE. ?BURY "PLAY.PACK
BUTFIRST BF	object	Outputs all but the first element in the input object. ?BUTFIRST "SMILES MILES ?BF [1 2 3] [2 3]
BUTLAST BL	object	Outputs all but the last element in the input object. ?BUTLAST [1 2 3 4] [1 2 3]
BYE		Exits current session of Logo and returns you to the Desktop. ?BYE
CATCH	name instr__list	Traps errors and special conditions that occur during the execution of the input instruction list. >CATCH "ERROR [DO.IT.UNTIL] >PRINT [I CAUGHT AN ERROR]

CHANGEF	new fname old fname	Change File. Changes the name of a file in the disk directory. ?CHANGEF "NEWFILE "OLDFILE
CHAR	n	Outputs the character whose ASCII value is the input number. ?CHAR 83 S
CIRCLE	[x y radius]	Makes Logo draw a circle at the x- and y- coordinates with the input radius.
CLEAN		Erases the viewport without affecting the turtle. ?CLEAN
CLEARSCREEN CS		Erases the viewport and puts the turtle at [0 0] heading 0 (North) with the pen down. ?CLEARSCREEN
CLEARTEXT CT		Erases all text in the text window then positions the cursor in the upper left-hand corner of the text window. ?CLEARTEXT
CO	<object>	COntinue. Ends a pause caused by PAUSE, a **[Control][Z]** keystroke, or ERRACT. TO SQUARE REPEAT 4 [FD 50 FT 90] END Pausing...in SQUARE: [PAUSE] SQUARE ?CO

COPYDEF	new__procname old__procname	Makes a copy of a procedure definition with another name within the computer's memory. ?COPYDEF "SQUARE "BOX
COPYOFF		Stops echoing text at the printer. ?COPYOFF
COPYON		Starts echoing text at the printer. ?COPYON
COS	degrees__n	COSine. Outputs the cosine of the input number of degrees. ?COS 60 .5
COUNT	object	Outputs the number of elements in the input object. ?COUNT "six 3 ?COUNT [0 1 2 3] 4
DEFINE	procname defin__list	Defines a new word—similar to TO but must be one line. ?DEFINE "SAY.Hi ?SAY.Hi Hi!
DEFINEDP	object	Outputs TRUE if the input name identifies a defined procedure. Otherwise it outputs FALSE. ?DEFINEDP "SAY.HI TRUE
DEGREES	radians__n	Outputs the number of degrees in the input number of radians. ?DEGREES 25 1432.394742

DIR	<fname>	Outputs a list of Logo file (.LOG) names on the default or specified disk and accepts an ambiguous file name. ?DIR [STARS.LOG] ?DIR "B: [B:AVERAGE.LOG B:TOOLS.LOG]
EDALL	<pkgname | pkgname_list>	Loads all the procedures and variables in the workspace or the specified package(s) into the text editor's buffer. ?EDALL
EDF	fname	Loads the specified disk file into the text editor's buffer or creates a new file. ?EDF "STARTUP
EDIT ED	<name | name_list>	Loads the specified procedure(s) and/or variable(s) into the text editor's buffer. ?ED "SQUARE
EDNS	<pkgname | pkgname list>	Loads all the variables in the workspace or the specified package(s) into the text editor's buffer. ?EDNS "PRACTICE.PACK
EDPS	<pkgname | pkgname list>	Loads all the procedures in the workspace or the specified package(s) into the text editor's buffer. ?EDPS "PLAY.PACK
ELLIPSE	[x y X-radius Y-radius]	Makes Logo draw an ellipse at the input x- and y- coordinates with the input X- and Y- radius.

EMPTYP	object	Outputs TRUE if the input object is an empty word or a empty list. Otherwise it outputs FALSE. ?EMPTYP '' TRUE ?EMPTYP [ ] TRUE ?EMPTYP [x] FALSE
END		Indicates the end of a procedure definition. END must be the last line of a procedure. ?TO SAY. HI >PRINT ''HI >END SAY.HI defined ?
EQUALP	object object	Outputs TRUE if input objects are equal numbers, identical words, or identical lists. Otherwise it outputs FALSE. ?EQUALP ''POP ''POP TRUE
ERALL	<pkgname | pkgname_list>	Erases all the unburied procedures and variables from the workspace or the specified unburied package(s). ?ERALL
ERASE ER	procname | procname_list	Erases the specified unburied procedure(s) from the workspace. ?ERASE ''BOX

ERASEFILE	fname	Erases the specified disk file. ?ERASEFILE "B:\PIGLATIN ?LOAD "B:\PIGLATIN File not found
ERN	varname | varname list	Erases the specified unburied variable(s) from the workspace. ?ERN [SIDE ANGLE]
ERNS	<pkgname | pkgname_list>	Erases all unburied variables from the workspace or the specified unburied package(s). ?ERN "DRAW.PACK
ERPS	<pkgname | pkgname_list>	Erases all unburied procedures from the workspace or the specified unburied package(s). ?ERPS [DRAW.PACK MOVE.PACK]
ERROR		Outputs a list whose elements describe the most recent error. ?ERROR [29 [Not enough input to CIRCLE] CIRCLE [CIRCLE] [ ] [ ]]
EXP	n	Outputs the natural exponent of the input number. ?EXP 1 2.71828
FENCE		Establishes a boundary that limits the turtle to plotting within the viewport. ?FENCE ?FORWARD 300 Turtle out of bounds.

FILL		Paints an area with the current fill color, changing the dot under the turtle (and all horizontally and vertically contiguous dots of the same color) to the current fill attributes. ?FILL
FILLATTR		Outputs the style, index, and color attributes of the current fill pattern.
FIRST	object	Outputs the first element of the input object. ?FIRST "ZEBRA Z ?FIRST [1 2 3] 1
FOLLOW	procname procname	Reorganizes the workspace so the first input-named procedure is followed by the second. FOLLOW does not change the order of procedures in a package definition. ?FOLLOW "FIRST "SECOND
FORWARD FD	distance_n	Moves turtle the input number of steps in the direction of its current heading. ?FORWARD 100
FPUT	object object	Outputs a new object formed by making the first input object the first element in the second input object. ?FPUT "S "MILES SMILES ?FPUT 1 [2 3] [1 2 3]

GETTEXT		Outputs the effect number of the current special graphic text attributes.
GLIST	prop <pkgname |pkgname_list>	Outputs a list of all objects in the workspace or specified package(s) that have the input property in their property lists. ?GLIST "\.DEF "FLY [FLY BUZZ ZOOM]
GO	word	Executes the line within the current procedure following a LABEL expression with the same input word. ?GO "LOOP
GPROP	name prop	Outputs the value of the named property of the named object. ?MAKE "HEIGHT "72 ?GPROP "HEIGHT ".APV 72 ?GPROP "HEIGHT ".DEV [ ]
HEADING		Outputs the number that indicates the turtle's current heading. ?HEADING 126
HIDETURTLE HT		Makes the turtle invisible, which speeds and clarifies the drawing. ?HIDETURTLE
HOME		Returns the turtle to position [0 0] heading 0 (North). ?HOME

IF	pred_exp instr_list <instr_list>	Executes one of two literal instruction lists depending on the value of the input predicate expression. >IF (:A > :B) [PRINT [:A IS BIGGER]] > [PRINT [:B IS BIGGER]]
IFFALSE IFF	instr list	Executes the input instruction list if the most recent TEST expression was FALSE. (See TEST explanation for example.)
IFTRUE IFT	instr_list	Executes the input instruction list if the most recent TEST expression was TRUE. (See TEST explanation for example.)
INT	n	Outputs the integer portion of the input number. ?INT 3.333 3
ITEM	n object	Outputs the specified element of the input object. ?ITEM 4 "DWARF R
KEYP		Outputs TRUE if a character has been typed at the keyboard and is waiting to be read. ?KEYP FALSE
LABEL	word	Identifies the line to be executed after a GO expression with the input word. LABEL "LOOP

LAST	object	Outputs the last element of the input object. ?LAST [0 2 4] 4
LEFT LT	degrees__n	Rotates the turtle the input number of degrees to the left. ?LEFT 90
LINEATTR		Outputs the style, width, and color attributes of the current line type.
LIST	object object (...)	Outputs a list made up of the input objects; retains the list's outer brackets. ?LIST "BIG [FEET] [BIG [FEET]] ?(LIST) ?(LIST 1 2 3 4) [1 2 3 4]
LISTP	object	Outputs TRUE if the input object is a list. Otherwise it outputs FALSE. ?LISTP "WORD FALSE
LOAD	fname <pkgname>	Reads the input-named Logo file (.LOG) from the disk into the workspace. ?LOAD "PIGLATIN "PIG.PACK BEGIN.VOWELP defined PIG defined PIGLATIN defined
LOADPIC	fname	Paints the graphic design saved in the input-named picture file onto the graphic viewport. ?LOADPIC "B:\DESIGNS

LOCAL	varname (...)	Makes the input-named variable(s) accessible only to the current procedure and the procedures it calls. ?(LOCAL "A "B "C)
LOG	n	Outputs the natural logarithm of the input number. ?LOG 2 0.693147
LOG10	n	Outputs the base 10 common logarithm of the input number. ?LOG10 100 2
LOWERCASE LC	word	Outputs the input word with all alphabetic characters in the lower case. ?LOWERCASE "SOUTH south
LPUT	object object	Outputs a new object formed by making the first input object the last element in the second input object. ?LPUT 4 [1 2 3] [1 2 3 4] LPUT "A [BCD] [BCDA]
MAKE	varname object	Outputs a new object formed by making the first input object the last element in the second input object. ?MAKE "SIDE 50 ?:SIDE 50

MEMBERP	object object	Outputs TRUE if the first input object is an element of the second input object. Otherwise it outputs FALSE. ?MEMBERP "Y "ONLY TRUE
MOUSE		Outputs a list that contains the current mouse state in the form [x y b1 b2 b3]. x and y are the coordinate positions b1 and b2 are left and right mouse buttons that output TRUE if pressed. b3 outputs TRUE if the mouse pointer is over the graphic viewport. Otherwise it outputs FALSE. ?MOUSE [50 35 TRUE FALSE TRUE]
NAME	object varname	Makes the input object the value of the input-named variable. ?NAME 50 "SIDE ?:SIDE 50
NAMEP	word	Outputs TRUE if the input word identifies a defined variable. Otherwise it outputs FALSE. ?MAKE "FLAVOR "CHOCOLATE ?NAMEP "FLAVOR TRUE ?NAMEP "VANILLA FALSE (When previously indicated)
NODES		Outputs the number of free nodes in the workspace (1 node = 4 bytes). ?NODES 684

NOFORMAT		Removes procedure formatting, including comments, from the workspace. ?NOFORMAT
NOT	exp	Outputs TRUE if the input expression is FALSE. Outputs FALSE if the input is TRUE. ?NOT (3 = 4) TRUE
NOTRACE		Turns off trace monitoring of procedure execution. ?NOTRACE
NOWATCH	<procname | procname_list>	Turns off watch monitoring of all or specified procedure(s). ?NOWATCH "AVERAGE
NUMBERP	object	Outputs TRUE if the input object is a number. Otherwise it outputs FALSE. ?NUMBERP "TWO FALSE ?NUMBERP "2 TRUE
OR	exp exp (...)	Outputs FALSE if all input expressions are FALSE. Otherwise it outputs TRUE. ?OR (1 = 1) (1 = 3) TRUE

OUTPUT OP	object	Makes the input object the output of the procedure and exits the procedure at that point. The following example outputs from within a procedure. >IF 24 = 4 * 6 [OUTPUT ''TRUE] TRUE
PACKAGE	pkgname name | name_list	Puts the name(s) into the input-named package. ?PACKAGE ''SIZES [BIG SMALL]
PALETTE PAL	color n	Outputs the RGB list for the specified color number. (See SETPAL to change colors.) ?PAL 1 [15 15 15]
PATH		Outputs the name of the current default drive and directory path. ?PATH A:
PAUSE		Suspends the execution of the current procedure to allow interaction with the interpreter or editor. ?IF :A > :P [PAUSE]
PENDOWN PD		Puts the turtle's pen down and the turtle resumes drawing. ?PENDOWN
PENERASE PE		Makes the turtle draw in the background color and the turtle erases the drawn lines. ?PENERASE

PENREVERSE PX		Makes the turtle change the color of any previously colored pixel in its trail to the reverse or logical color complement. ?PENREVERSE
PENUP PU		Picks the turtle's pen up and the turtle stops drawing. ?PENUP
PI		Outputs the value of PI: 3.1416
PIECE	n n object	Outputs an object that contains the specified elements of the input object. ?PIECE 2 4 [a b c d e] [b c d]
PKGALL	pkgname	Puts all procedures and variables not already in packages into the specified package. ?PKGALL ''OTHER
PLIST	name	Outputs the property list of the input-named object. ?MAKE ''BIRD ''BLUE ?PLIST ''BIRD [.APV BLUE]
PO	name | name__list	Displays the definition(s) of the specified procedure(s) or variable(s). ?PO ''X X is 5

POALL	<pkgname | pkgname_list>	Displays the definitions of all procedures and variables in the workspace or the specified package(s). ?POALL "PLAY.PACK
POCALL	procname	Displays the names of the procedures called by the input-named procedure. ?POCALL "AVERAGE AVERAGE ADDUP
POLY	(x1 y1 x2 y2 ...Xn Yn]	Makes Logo draw a polygon to input x- and y- coordinates.
PONS	<pkgname | pkgname_list>	Displays the names and values of all variables in the workspace or the specified package(s). ?PONS
POPKG	<pkgname | pkgname_list>	Displays the name and contents of each package in the workspace or the specified package(s). ?POPKG
POPS	<pkgname | pkgname_list>	Displays the names and definitions of all procedures in the workspace or the specified package(s). ?POPS
POREF	procname | procname_list	Displays the names of the procedures that call the input-named procedure(s). In the following example, triangle is a procedure within FLAG. ?POREF :"TRIANGLE TO FLAG

POS		Outputs a coordinate list of the turtle's current position. ?POS [90 22]
POTL		Displays the names of the TOP-LEVEL procedures. These procedures are not called by any other procedure in the workspace. ?POTL TO AVERAGE :NUMBERS
POTS	<pkgname | pkgname_list>	Displays the names and inputs of all procedures in the workspace or the specified package(s). ?POTS "SHAPES TO POLY :SIDE :ANGLE TO SPI :SIDE :ANGLE :INC.
PPROP	name propname prop val	Puts the input property pair into the name's property list. ?PPROP "KATHY "EXTENSION 82
PPS	<pkgname | pkgname_list>	Displays the non-system property pairs of all objects in the workspace or the specified package(s). ?PPS KATHY's EXTENSION is 82
PRIMITIVEP	object	Outputs TRUE if the input object is a primitive name. Otherwise it outputs FALSE. ?PRIMITIVEP "TEST TRUE

PRINT PR	object (...)	Displays the input object(s) on text window, file, or device. PRINT removes lists' outer brackets and follows last input with a carriage return. (Compare with SHOW and TYPE.) ?PRINT [A B C] A B C
PROCLIST		Outputs a list that contains the names of all defined procedures. ?PROCLIST [SQUARE AVERAGE ADDUP]
PRODUCT	n n (...)	Outputs the product of the input numbers. ?PRODUCT 2 2 4
QUOTIENT	n n	Outputs the integer quotient of the two input numbers and truncates the input numbers to integers before dividing. ?QUOTIENT 21 7 3
RADIANS	degrees_n	Outputs the number of radians in the input number of degrees. ?RADIANS 90 1.570796
RANDOM	n	Outputs a random integer. The input number must be between 32767 and -32768. ?RANDOM 20 19

READCHAR
RC

Outputs the first character typed at the keyboard or entered from a file or device.

```
?MAKE "KEY READCHAR
?:KEY
R
```

READLIST
RL

Outputs a list that contains a line typed at the keyboard (input must be followed by a carriage return) or read from a data file.

```
?READLIST
1 2 3
[1 2 3]
```

READQUOTE
RQ

Outputs a word that contains a line typed at the keyboard or read from a data file. READ-QUOTE input must be followed by a carriage return.

```
?READQUOTE
1 2 3
1 2 3
```

RECYCLE

Frees as many nodes as possible and reorganizes the workspace.

```
?RECYCLE
```

REMAINDER n n

Outputs the integer remainder obtained when the first input number is divided by the second.

```
?REMAINDER 7 3
1
```

REMPROP	name prop	Removes the specified property from the name's property list. ?MAKE "PACK "Color ?PONS ?Pack IS COLOR ?REMPROP ?PONS ?
REPEAT	n instr_list	Executes the input instruction list the input number of times. ?REPEAT 4 [FORWARD 50 RIGHT 90]
RERANDOM		Makes a subsequent RANDOM or SHUFFLE expression reproduce the same random sequence. ?RERANDOM ?RANDOM 20 19 ?RERANDOM ?RANDOM 20 19
RIGHT RT	degrees_n	Rotates the turtle the input number of degrees to the right. ?RIGHT 45
ROUND	n	Outputs the input number rounded off to the nearest integer. ?ROUND 3.333 3
RUN	instr list	Executes the input instruction list. ?RUN [PRINT "HI] HI

SAVE	fname <pkgname | pkgname_list	Writes the contents of the workspace or specified package(s) to the input named disk file. If the name is less than nine characters, .LOG is added to the filename. If (.) is used after the first character only, the next three letters are used as filename extensions. ?SAVE "MYFILE
SAVEPIC	fname	Writes the contents of the graphic viewport to the input named picture file. ?SAVEPIC "DESIGN3
SCREENFACTS SF		Outputs a list that describes the graphic viewport's attributes. The format is: [BGCOLOR VIEWPORT-MODE SCRUNCH ZOOM XPAN YPAN] BGCOLOR = Background color number of graphic viewport. VIEWPORT-MODE = WINDOW, WRAP, or FENCE mode. SCRUNCH = Current aspect ratio of the graphic viewport. ZOOM = Magnification factor for the visible objects on the graphic viewport. XPAN YPAN = Center point of the viewport in the graphic plane. ?SETBG 2 ?WINDOW SETSCRUNCH 2 SETZOOM 2 ?SETPAN [100 100] ?SCREENFACTS [2 WINDOW 2 2 100 100]

SENTENCE SE	object object (...)	Outputs a list made up of the input objects and removes the lists' outer brackets. ?SENTENCE "HARE [RABBIT BUNNY] [HARE RABBIT BUNNY]
SETBG	color__n	Sets the graphic viewport background to the color represented by the input number. CLEARSCREEN must follow to display new background color. ?SETBG 1
SETFILL	[style__n index__n color__n]	Sets the fill pattern to the input numbered style, index, and color.
SETHEADING SETH	degrees__n	Turns the turtle to the absolute heading specified by the input number of degrees. The positive numbers turn the turtle clockwise; negative numbers counter-clockwise. To point the turtle East, enter the following: ?SETHEADING 90
SETLINE	[style__n width__n color__n]	Sets the line type to the input numbered style, width, and color.
SETPAL	color n RGB__list	Sets the input color number to the color combination of the input RGB__list values. PAL 1 [0 0 0] SETPAL 1 [1000 0 0] PAL 1 [1000 0 0]

SETPAN	coord list	Establishes the center point of the viewport in the turtle plane. Default is [0 0]. SETPAN doesn't clear the viewport nor alter any previous drawing. REPEAT Y [FD 50 RT 50] ?SETPAN [50 50] REPEAT Y [FD 50 RT 50] SETPAN [0 0]
SETPATH	d:	Makes the specified pathname the default pathname. Used to change disk drives. Access: A for startup drive,B for second drive. ?SETPATH ''B:\PATHNAME
SETPC	color__n	Sets the turtle's pen to the color specified by the input color number. TO PENCOL MAKE ''N RANDOM 2 FD 25 RT 22.5 SETPC :N PENCOL END
SETPEN	list	Sets the turtle's pen to the state and color specified in the input list. ?SETPEN [PD 2]
SETPOS	coord list	Moves the turtle to the position specified in the input coordinate list. SETPOS [50 50]

SETSCRUNCH	n	Sets the graphic viewport's vertical aspect ratio to the input number. SETSCRUNCH doesn't clear the viewport nor alter anything previously drawn. ?SETSCRUNCH .5
SETTEXT	effect_n	Sets a special attribute for the graphic text to the input numbered effect.
SETX	n	Moves the turtle horizontally to the x coordinate specified by the input number. ?SETX −50
SETY	n	Moves the turtle vertically to the y coordinate specified by the input number. ?SETY 90
SETZOOM	n	Allows you to zoom in or out to magnify your graphic displays. SETZOOM doesn't clear the viewport nor alter any previous drawing. ?SETZOOM 2
SHOW	object	Outputs the input object on the text window, data file, or system device. SHOW retains the list's outer brackets and follows the input with a carriage return. (Compare with PRINT and TYPE.) ?SHOW [A B C] [A B C]
SHOWTURTLE ST		Makes the turtle visible if hidden. ?SHOWTURTLE

SHUFFLE	list	Outputs a list that contains the elements of the input list in random order. ?SHUFFLE [1 2 3 4] [3 2 4 1]
SIN	degrees_n	Outputs the sine of the input numbert of degrees. ?SIN 30 .5 ?Degrees SIN 30 28.647892
SORT	list	Outputs a list of input words sorted into ascending order. ?SORT[D C B A 4 3 2 1] [1234ABCD]
SQRT	n	Outputs the square root of the input number. ?SQRT 25 5
STOP		Stops the execution of the current procedure and returns to TOPLEVEL (the ? prompt) or the calling procedure.
SUM	n n (...)	Outputs the sum of the input numbers. ?SUM 2 2 4
TAN	degrees_n	Outputs the tangent of the specified angle. ?TAN 45 1

TEST	exp	Remembers whether the input expression is TRUE or FALSE for subsequent IFFALSE or IFTRUE expressions. ?TO FLIP.COIN >TEST 1 = RANDOM 2 >IF 1 = RANDOM 100 > [PRINT [LANDED ON EDGE] STOP] >IFTRUE [TYPE "HEADS] >IFFALSE [TYPE "TAILS] >PRINT [\ SIDE UP.] >END FLIP.COIN defined ?
TEXT	procname	Outputs the definition list of the specified procedure. ?TEXT "SQUARE [[ ] [REPEAT 4 [FORWARD 50 RIGHT 90]]]
THING	varname	Outputs the value of the input-named variable. ?MAKE "CHOCOLATE "SEMI#-SWEET ?THING "CHOCOLATE SEMI-SWEET
THROW	name	Executes the line identified by the input name in a previous CATCH expression. >IF :A < :B [THROW "BIGGER]
TO	procname <inputs>	Indicates the beginning of a procedure definition. ?TO SQUARE :SIDE >REPEAT 4 [FD :SIDE RIGHT 90] >END SQUARE defined ?

TOWARDS coord__list

Outputs a heading that makes the turtle face the position specified in the input coordinate list.

```
?FORWARD 50 RIGHT 90
?TOWARDS [0 0]
180
```

TRACE

Turns on trace monitoring of procedure execution and variable assignment. TRACE displays the name of each procedure as it is called and the name and value of each variable as it is defined. TRACE allows observation of the procedure's execution without interruption.

```
?TRACE
```

TURTLEFACTS
TF

Outputs a list that describes the turtle's attributes. The format is:

```
[XCOR YCOR HEADING
PENSTATE PENCOLOR__N
SHOWNP]
```

XCOR = Turtle's x coordinate.
YCOR = Turtle's y coordinate.
HEADING = Compass direction the turtle is facing.
PENSTATE = PD for pendown, PE for penerase, PX for penreverse, or PU for penup.
PENCOLOR = Pen's color number.
SHOWNP = TRUE if the turtle is visible.

```
?SETPOS [15 30] RIGHT 60
?PENERASE SETPC 3
HIDETURTLE
?TURTLEFACTS
[15 30 60 PE 3 FALSE]
```

TURTLETEXT TT	object (...)	Displays the input object(s) at the turtle's current location on the graphic viewport in the current pen color and state. ?TURTLETEXT "HI
TYPE	object (...)	Outputs the input object(s) on the text window, data file, or system device. TYPE removes the lists' outer brackets but does not follow the last input with a carriage return. (Compare with PRINT and SHOW.) ?TYPE [A B C] A B C
UNBURY	pkgname |	Restores the specified package(s) to workspace management commands. ?UNBURY "PLAY.PACK
UPPERCASE	word	Outputs the input word with all alphabetic characters in the uppercase mode. ?UPPERCASE "Jones JONES
WATCH	<procname | procname_list>	Turns on the expression-by-expression procedure execution monitor. WATCH allows interaction with the interpreter or editor. ?WATCH "AVERAGE

WHERE		Outputs the item number of the most recent successful MEMBERP expression. ?MEMBERP "R [Q R S] TRUE ?WHERE 2
WINDOW		Allows the turtle to plot outside the viewport after a WRAP or FENCE expression. ?FENCE FD 300 [RETURN] ?WINDOW FD 300 ?CS
WORD	word word (...)	Outputs a word made up of the input words. ?WORD "SUN "SHINE SUNSHINE
WORDP	object	Outputs TRUE if the input object is a word or a number. Otherwise it outputs FALSE. ?WORDP "HI TRUE ?WORDP [HI] FALSE
WRAP		Makes the turtle reappear on the opposite side of the graphics window when it exceeds the boundary. ?WRAP
XCOR		Outputs the x coordinate of the turtle's current position. ?XCOR 145

YCOR		Outputs the y coordinate of the turtle's current position. ?YCOR 36
+	a b (...)	Infix or prefix primitive and delimiter. Outputs the sum of the input numbers. ?2 + 2 4
–	a b	Infix or prefix primitive and delimiter. Outputs the difference of the two input numbers. ?10 - 5 5
*	a b (...)	Infix or prefix primitive and delimiter. Outputs the product of the input numbers. ?4 * 6 24
/	a b	Infix or prefix primitive and delimiter. Outputs the decimal quotient of the two input numbers. ?25/5 5
^	a b	Infix or prefix primitive and delimiter. Outputs the exponent of the two input numbers. ?10 ^ 2 (read 10 to the 2nd) 99.999961

<	a b	Infix or prefix primitive and delimiter. Outputs TRUE if the first input word is less than the second. Otherwise it outputs FALSE. ?13 < 27 TRUE
>	a b	Infix or prefix primitive and delimiter. Outputs TRUE if the first input word is greater than the second. Otherwise it puts FALSE. ?20 > 19 TRUE
=	a b	Infix or prefix primitive and delimiter. Outputs TRUE if the two input objects are equal. Otherwise it outputs FALSE. ?1 = 2 FALSE ?"logo = "logo TRUE
< >	a b	Infix or prefix primitives and delimiters. Outputs TRUE if the two objects are not equal. Otherwise it outputs FALSE. ?< >1 2 TRUE ?< >2 2 FALSE

><	a b	Infix or prefix primitives and delimiters. Outputs TRUE if the two objects are not equal to each other. Otherwise it outputs FALSE. ?>< 3 1 TRUE ?>< 3 3 FALSE
>=	a b	Infix or prefix primitives and delimiters. Outputs TRUE if the first word is greater than or equal to the second. Otherwise it outputs FALSE. ?>= 3 4 FALSE ?>= 3 3 TRUE ?>= 5 3 TRUE
=>	a b	Infix or prefix primitives and delimiters. Outputs TRUE if the first word is greater than or equal to the second. Otherwise it outputs FALSE. ?=> 13 15 FALSE ?=> 54 54 TRUE ?=> 75 3 TRUE

< =	a b	Infix or prefix primitives and delimiters. Outputs TRUE if the first word is less than or equal to the second. Otherwise it outputs FALSE. ?< =4 7 TRUE ?< =4 4 TRUE ?< =7 4 FALSE
= <	a b	Infix or prefix primitives and delimiters. Outputs TRUE if the first word is less than or equal to the second. Otherwise it outputs FALSE. ?= <14 18 TRUE ?= <40 40 TRUE ?= <87 4 FALSE

ATARI Logo Special Characters

;	Delimiter. Indicates comments to be ignored by the interpreter.
(	Delimiter. Begins an enclosed expression that contains multiple inputs or groups of numeric expressions and specifies the order of operations.
)	Delimiter. Ends an enclosed expression that contains multiple inputs or groups of numeric expressions and specifies the order of operations.

''	Forces Logo to interpret a word as an object instead of as a procedure name.
[ ]	Delimiters. Enclose elements of a list.
#	Forces Logo to interpret a special character as a literal character.

APPENDIX G
FUNCTIONAL COMMAND LIST

The primitives are grouped by function and the input form is indicated where applicable.

Arithmetic Operations

ABS n
ARCTAN n
COS degrees_n
DEGREES radians_n
EXP n
INT n
LOG n
LOG10 n
PI
PRODUCT n n (...)
QUOTIENT n n
RADIANS degrees_n
RANDOM n
REMAINDER n n
RERANDOM
ROUND n
SIN degrees_n
SQRT n
SUM n n (...)
TAN degrees_n
+ a b (...)
- a b
* a b (...)
/ a b
^ a b

Conditionals and Flow of Control

BYE
CO <object>
GO word
IF pred_exp instr_list <instr_list>
IFFALSE, IFF instr_list
IFTRUE, IFT instr_list
LABEL word
OUTPUT, OP object
REPEAT n instr_list
RUN instr_list
STOP
TEST pred_exp

Defining Procedures

COPYDEF new_procname old_procname
DEFINE procname defin_list
DEFINEDP object
PRIMITIVEP object
TEXT procname

Defining Variables

LOCAL varname (...)
MAKE varname object
NAME object varname
NAMEP word
THING varname

Disks

PATH
SETPATH d:

Editing Procedures and Variables

EDALL pkgname |pkgname_list>
EDIT, ED <name | name list>
EDNS pkgname | pkgname_list>
EDPS <pkgname | pkgname_list>

Error Handling and Debugging

CATCH name instr__list
ERROR
NOTRACE
NOWATCH <procname | procname__list>
PAUSE
THROW name
TRACE
WATCH <procname | procname__list>

Files

CHANGEF new__fname old__fname
DIR <fname>
EDF fname
ERASEFILE fname
LOAD fname <pkgname>
SAVE fname <pkgname | pkgname__list>

Graphic Movement

ARC [x y radius begin__angle end__angle
BACK, BK distance__n
BOX [x y width height]
CIRCLE [x y radius]
ELLIPSE [x y X-radius Y-radius]
FORWARD, FD distance__n
HEADING
HIDETURTLE HT
HOME
LEFT, LT degrees__n
POLY [x1 y1 x2 y2...xn yn]
POS
RIGHT, RT degrees__n
SETHEADING, SETH degrees__n
SETPOS coord__list
SETX n
SETY n
SHOWTURTLE, ST
TOWARDS coord__list
XCOR
YCOR

Graphic Viewport

CLEAN
CLEARSREEN, CS
FENCE
FILL
FILLATTR
GETTEXT
LINEATTR
LOADPIC fname
PAL color__n
PENDOWN, PD
PENERASE, PE
PENREVERSE, PX
PENUP, PU
SAVEPIC fname
SETBG color__n
SETFILL [style__n index__n color__n]
SETLINE [style__n width__n color__n]
SETPC color__n
SETPEN list
SETPAL color__n RGB list
SETPAN coord__list
SETSCRUNCH n
SETTEXT effect__n
SETZOOM n
SCREENFACTS, SF
TURTLEFACTS, TF
TURTLETEXT, TT object
WINDOW
WRAP

Keyboard

KEYP
READCHAR, RC
READLIST, RL
READQUOTE, RQ

Logical Operations

AND pred_exp pred_exp (...)
NOT pred_exp
OR pred_exp pred_exp (...)
= a b
< a b
> a b
<> or >< a b
>= or =< a b
<= or => a b

Peripheral Devices

COPYOFF
COPYON
MOUSE [x y b1 b2 b3]

Property Lists

GLIST prop <pkgname | pkgname_list>
GPROP name prop
PLIST name
PPROP name prop object
PPS <pkgname | pkgname_list>
REMPROP name prop

Text Window

CLEARTEXT, CT
PRINT, PR object (...)
SHOW object
TYPE object (...)

Word and List Processing

ASCII word
BUTFIRST, BF object
BUTLAST, BL object
CHAR n
COUNT object
EMPTYP object
EQUALP object object
FIRST object

FPUT object object
ITEM n object
LAST object
LIST object object (...)
LISTP object
LOWERCASE, LC word
LPUT object object
MEMBERP object object
NUMBERP object
PIECE n n object
PROCLIST
SENTENCE, SE object object (...)
SHUFFLE list
SORT list
UPPERCASE, UC word
WHERE
WORD word word (...)
WORDP object

Workspace Management

BURY pkgname | pkgname_list
ERALL pkgname | pkgname_list>
ERASE, ER procname | procname_list
ERN varname | varname_list
ERNS <pkgname | pkgname_list>
ERPS <pkgname | pkgname_list>
FOLLOW procname procname
NODES
NOFORMAT
PACKAGE pkgname name | name_list
PKGALL pkgname
PO name | name_list
POALL <pkgname | pkgname_list>
POCALL procname
PONS <pkgname |pkgname_list>
POPKG <pkgname |pkgname_list>
POPS <pkgname | pkgname_list>
POREF procname | procname_list
POTL
POTS <pkgname | pkgname_list>
RECYCLE
UNBURY pkgname | pkgname_list

APPENDIX H
ST ASCII CHARACTER SET

The following tables show the complete character sets available on the ST Computer. To print any of these characters from ATARI Logo, type:

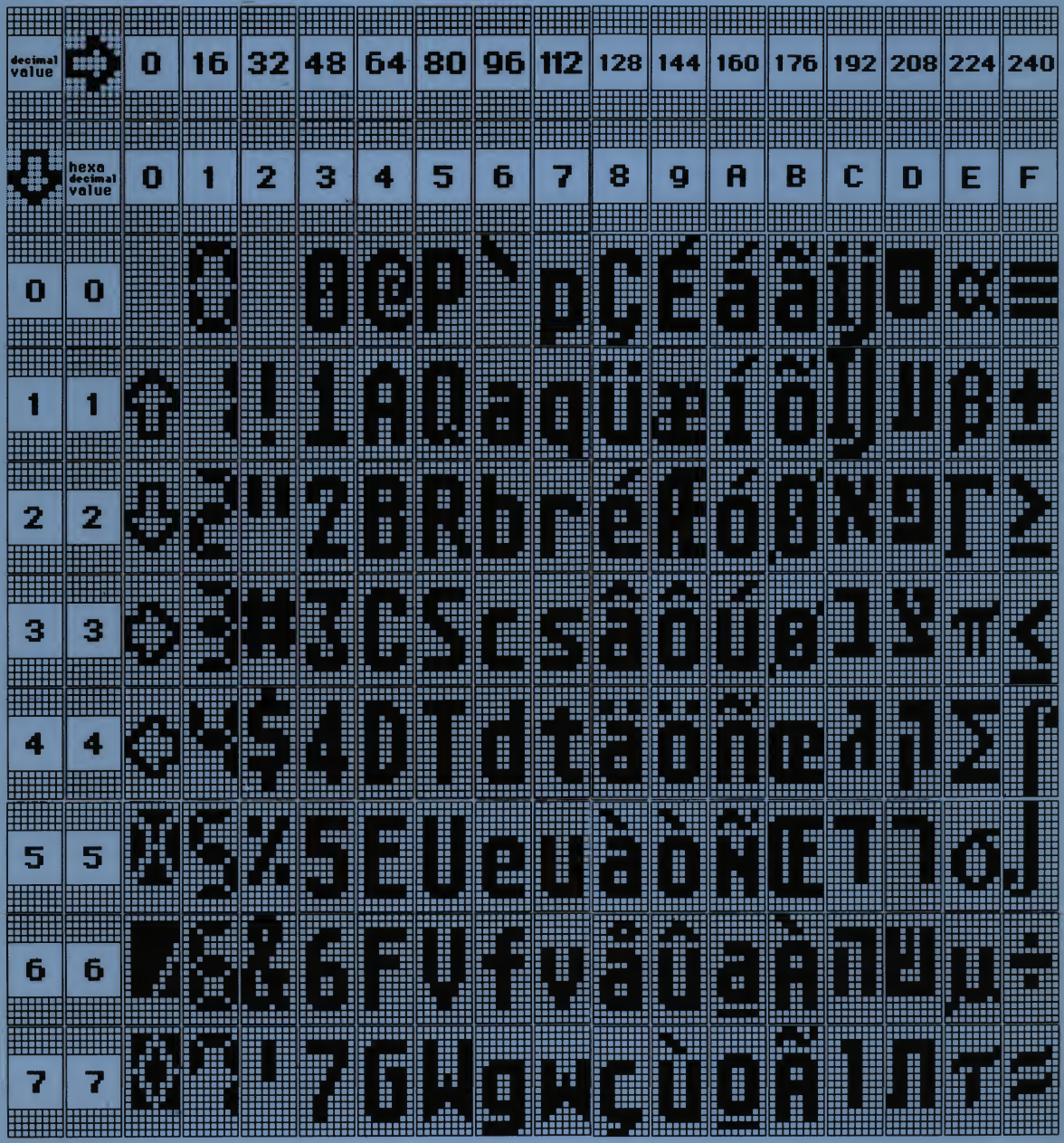
?PRINT CHAR n

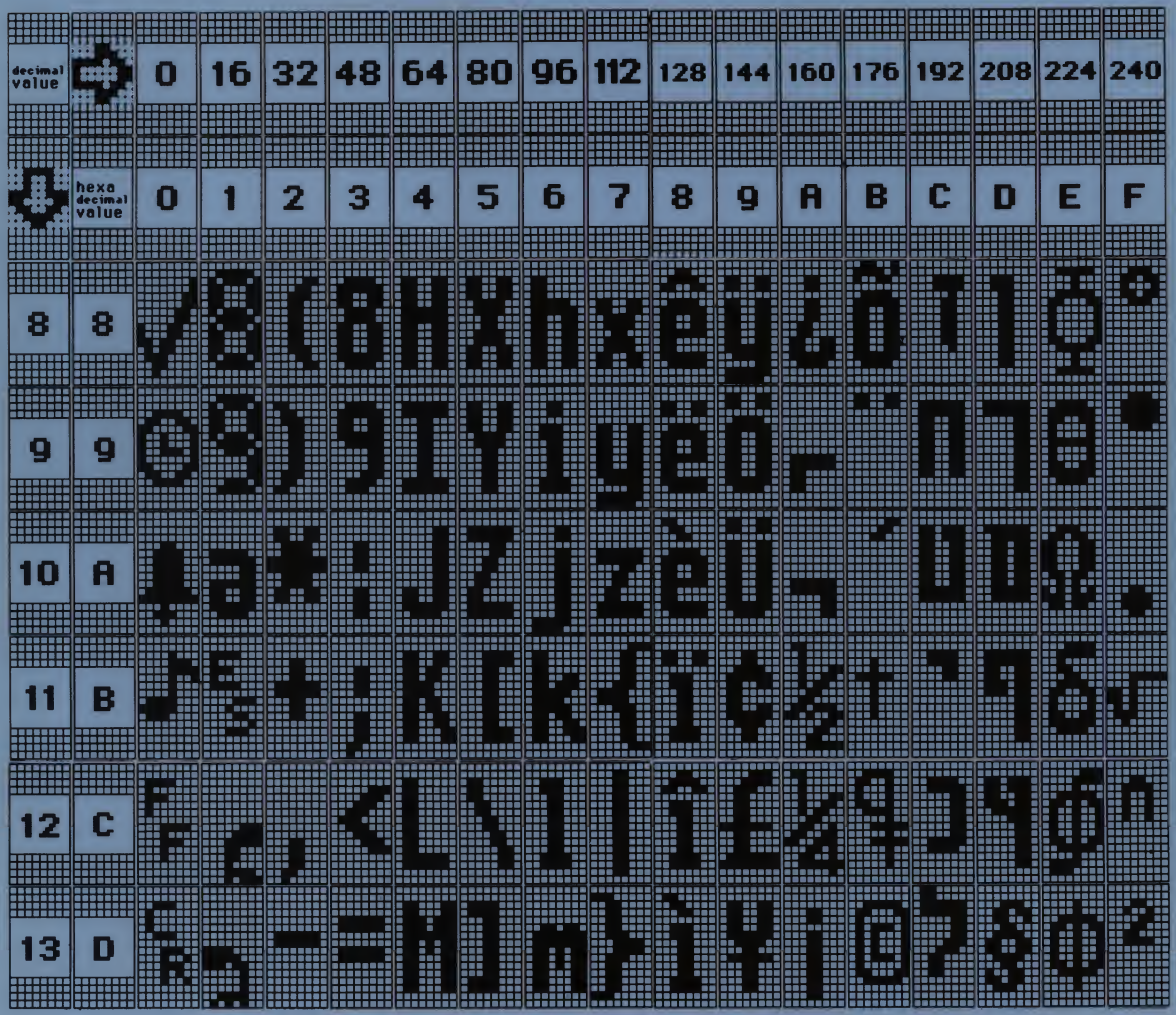
Replace the letter n with the ASCII value from the table.

There are two character tables. The first is set up for 8 × 8 characters; the second for 8 × 16 characters. The different character set sizes are used with different screen resolutions.

decimal value		0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
hexa decimal value		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0																
1	1																
2	2																
3	3																
4	4																
5	5																
6	6																
7	7																

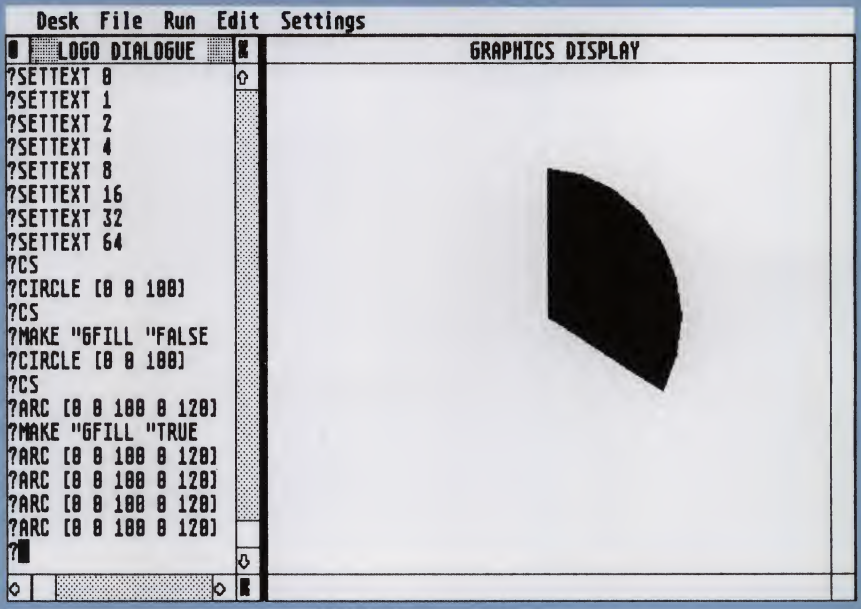

decimal value
hexa decimal value
0 16 32 48 64 80 96 112 128 144 160 176 192 208 224 240
0 1 2 3 4 5 6 7 8 9 A B C D E F
8 8
9 9
10 A
11 B
12 C
13 D
14 E
15 F

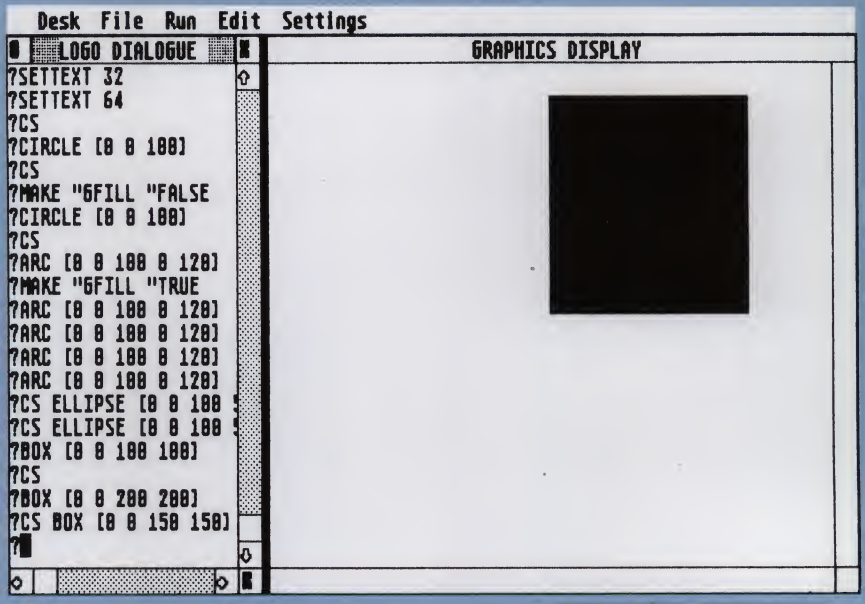

decimal value
0 16 32 48 64 80 96 112 128 144 160 176 192 208 224 240
hexa decimal value
0 1 2 3 4 5 6 7 8 9 A B C D E F
0 0
1 1
2 2
3 3
4 4
5 5
6 6
7 7

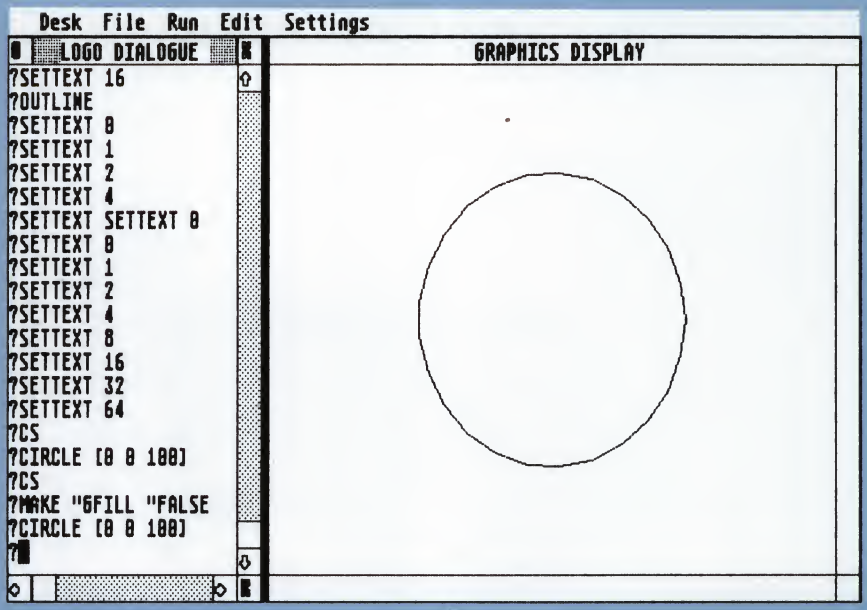

decimal value
0 16 32 48 64 80 96 112 128 144 160 176 192 208 224 240
hexa decimal value
0 1 2 3 4 5 6 7 8 9 A B C D E F
8 8
9 9
10 A
11 B
12 C
13 D

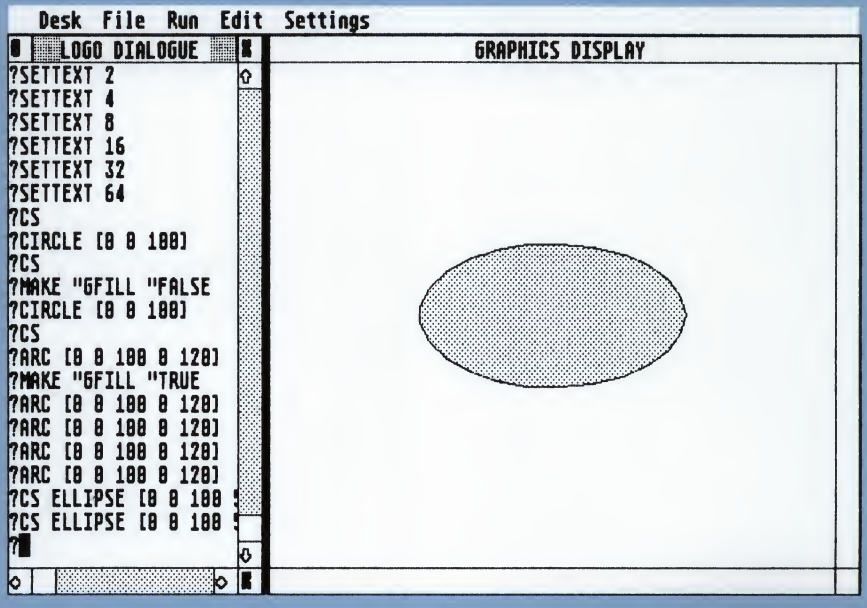

decimal value
0
16
32
48
64
80
96
112
128
144
160
176
192
208
224
240
hexa decimal value
0
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F
14
E
15
F

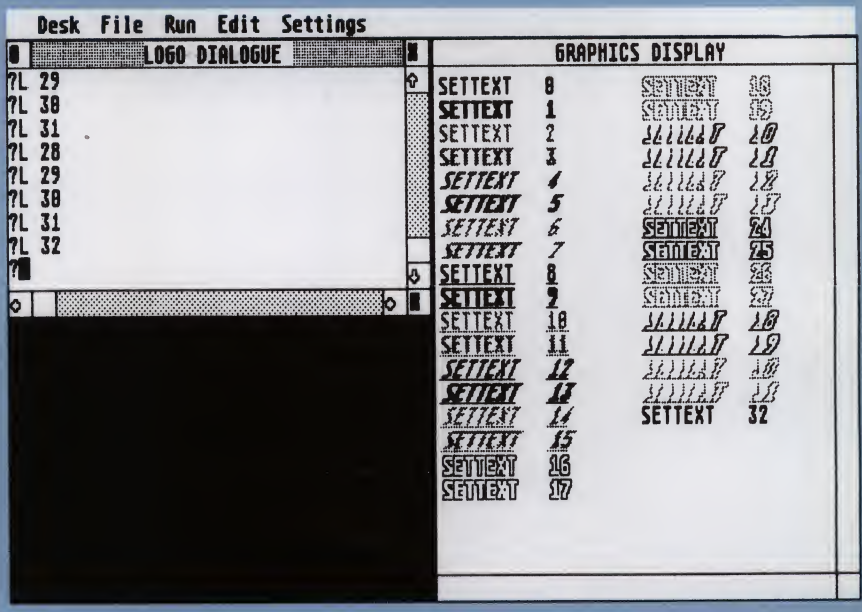
APPENDIX I
SAMPLE LOGO PROCEDURES






Desk File Run Edit Settings
LOGO DIALOGUE
?SETTEXT 16
?OUTLINE
?SETTEXT 0
?SETTEXT 1
?SETTEXT 2
?SETTEXT 4
?SETTEXT SETTEXT 0
?SETTEXT 0
?SETTEXT 1
?SETTEXT 2
?SETTEXT 4
?SETTEXT 8
?SETTEXT 16
?SETTEXT 32
?SETTEXT 64
?CS
?CIRCLE [0 0 100]
?CS
?MAKE "GFILL "FALSE
?CIRCLE [0 0 100]
?
GRAPHICS DISPLAY


Desk File Run Edit Settings
LOGO DIALOGUE
?SETTEXT 2
?SETTEXT 4
?SETTEXT 8
?SETTEXT 16
?SETTEXT 32
?SETTEXT 64
?CS
?CIRCLE [0 0 100]
?CS
?MAKE "GFILL "FALSE
?CIRCLE [0 0 100]
?CS
?ARC [0 0 100 0 120]
?MAKE "GFILL "TRUE
?ARC [0 0 100 0 120]
?ARC [0 0 100 0 120]
?ARC [0 0 100 0 120]
?ARC [0 0 100 0 120]
?CS ELLIPSE [0 0 100
?CS ELLIPSE [0 0 100
?
GRAPHICS DISPLAY


Desk File Run Edit Settings
LOGO DIALOGUE
?L 29
?L 30
?L 31
?L 28
?L 29
?L 30
?L 31
?L 32
?
GRAPHICS DISPLAY
SETTEXT 0
SETTEXT 1
SETTEXT 2
SETTEXT 3
SETTEXT 4
SETTEXT 5
SETTEXT 6
SETTEXT 7
SETTEXT 8
SETTEXT 9
SETTEXT 10
SETTEXT 11
SETTEXT 12
SETTEXT 13
SETTEXT 14
SETTEXT 15
SETTEXT 16
SETTEXT 17
SETTEXT 32

CUSTOMER SUPPORT

Atari Corp. welcomes any questions you might have about your ATARI Computer product.

Write to:

Atari Customer Relations
P.O. Box 61657
Sunnyvale, CA 94088

Please write the subject of your letter on the outside of the envelope.

We suggest that you contact your local Atari user groups. They are outstanding sources of information on how to get the most out of your ATARI Computer. To receive a list of user groups in your area, send a self-addressed stamped envelope to:

Atari User Group List
P.O. Box 61657
Sunnyvale, CA 94088

ATARI® ST LOGO™ ADDENDUM

ST Logo has been expanded and improved since the first release of the language. Six new primitives have been added: CALL, SOUND, FULLSCREEN, SYSFACTS, .WDEPOSIT, and .WEXAMINE.

The information on this Addendum should be used in conjunction with your *ATARI Logo Sourcebook*. Each new primitive is explained. A detailed discussion of the SOUND primitive is included for the advanced programmer.

Note: The name of the language has been changed from ATARI Logo to ST Logo. All references to the language in this Addendum will use the new name.

NEW ST LOGO PRIMITIVE SUMMARY

CALL

Syntax: CALL X

Description: Executes an assembly language routine located at location X. A good area to place assembly language programs is right after the screen buffer. This location can be calculated by adding 27,000 to the location of the screen buffer given in SYSFACTS. There are about 5,700 free bytes at this location. For example:

```
TO BELL
MAKE "ADDR ITEM 6 SYSFACTS
MAKE "ADDR :ADDR +27000
.WDEPOSIT :ADDR 16188
.WDEPOSIT :ADDR +2 7
.WDEPOSIT :ADDR +4 16188
.WDEPOSIT :ADDR +6 2
.WDEPOSIT :ADDR +8 16188
.WDEPOSIT :ADDR +10 3
.WDEPOSIT :ADDR +12 20045
.WDEPOSIT :ADDR +14 23631
.WDEPOSIT :ADDR +16 20085
CALL :ADDR
END
```

SOUND	Syntax:	SOUND [x1 x2 x3 x4 . . .]
	Description:	ST Logo takes the given list and interprets it as a set of primitives to be passed on to the ST sound chip.
FULLSCREEN (FS)	Syntax:	FULLSCREEN or FS
	Description:	Opens Graphics Display window to the largest possible size.
SYSFACTS	Syntax:	SYSFACTS
	Description:	Returns a list containing the following information: Graphics window's working x, y, w, and h; the maximum color index; the address of the Graphics buffer; the initial aspect ratio that ST Logo booted up with.
.WDEPOSIT	Syntax:	.WDEPOSIT X Y
	Description:	Places the 16 bit value of Y into the address specified by X. X must be an even address.
.WEXAMINE	Syntax:	.WEXAMINE X
	Description:	Examines the 16 bit contents of the address specified by X. X must be an even address.

ST LOGO SOUND PRIMITIVE

The SOUND primitive is a hook into the Bios Dosound routine. The function is invoked by typing SOUND followed by a list of numbers whose elements are in the range of 0-255, e.g., SOUND [x1 x2 x3 x4 x5 . . .]. This list represents a set of commands for the sound chip to execute. The ST Computer's sound chip uses 16 registers which determine what sound is produced. Each of the following commands takes parameters which are then placed into these registers.

Command numbers 0-15 take one argument to be placed into the appropriate register. For instance, SOUND [0 1] places a 1 into register 0, SOUND [0 1 1 2] puts a 1 into register 0 and places a 2 into register 1.

Command 128 takes one argument which is placed into a temporary location. This command works together with command 129.

Command 129 takes three arguments. The first argument is a register number to load, placing into this register the contents of the temporary value defined in command 128. The second argument is a value that is added to the temporary register. The third argument represents the termination value. This instruction is executed until the temporary register equals the termination value.

Commands 130-255 take one argument. If the argument is 0, the sound is terminated. Otherwise the argument reflects the amount of time that passes until the next command is executed. An argument of 50 will last for 1 second, 25 for ½ second, and so on (see example 3 at the top of page vi).

Example

To play TUNE3, enter the following procedures:

```
TO NOTE :FINE :COARSE :TIME
SOUND (LIST 0 :FINE 1 :COARSE 7 62 8 8 130 :TIME 7 63 255 0)
END
```

(Notice that any time a variable (:var) is included in any procedure, the brackets ([ ]) are changed to parentheses (( )) and the arguments are indicated as a LIST.)

```
TO CHORD :FINE1 :COARSE1 :FINE2 :COARSE2 :FINE3
:COARSE3 :TIME
SOUND (LIST 0 :FINE1 1 :COARSE1 2 :FINE2 3 :COARSE2 4
:FINE3 5 :COARSE3 7 56 8 8 9 8 10 8 130 :TIME 7 63 255 0)
END

TO TUNE3
MAKE "W 100 ; WHOLE NOTE
MAKE "H 50 ; HALF NOTE
MAKE "Q 25 ; QUARTER NOTE
MAKE "E 12.5 ; EIGHTH NOTE
MAKE "S 6.25 ; SIXTEENTH NOTE
NOTE 239 0 :E ; FIRST BAR
NOTE 63 1 :S
NOTE 63 1 :S
NOTE 123 1 :S
NOTE 222 1 :S
CHORD 126 2 250 1 102 1 :Q
CHORD 239 0 123 1 63 1 :E
END
```

SOUND REGISTERS

Use of the 16 sound chip registers is illustrated in the chart at the end of the Addendum.

Registers 0-5

Registers 0-5 control the tone produced by the three voices. They are paired off so that one register controls the coarse tuning of the voice's tone period while the other register controls the fine tuning. The chart illustrates the values to load into these registers to produce four octaves of notes.

Register 6

Besides the ability to generate tones, each voice is also capable of generating "noise." The period of this noise is controlled through register 6. A value of range 0-31 can be loaded into this register.

Register 7

Register 7 enables the three voices to output a tone or to output noise depending on which bits of the register are set. The selected voice can be instructed to generate a tone or noise by setting the respective bit to 0. The following table shows what value to place into this register to enable the desired voices:

TONES		NOISE	
Voice	Value	Voice	Value
A	62	A	55
B	61	B	47
C	59	C	31
A, B	60	A, B	39
B, C	57	B, C	15
A, C	58	A, C	23
A, B, C	56	A, B, C	7

Registers 8-10

These registers determine the volume of each of the three voices. Volume for each of the voices can range from 0-15 with 15 being the loudest. However, if bit 4, the "M" bit in the diagram, is set to a 1 (i.e., a value of 16) then the volume information is taken from registers 11, 12, and 13 (the envelope registers).

Registers 11-13

Registers 11-13 control the envelope shape that is generated. This envelope shape is formed by four stages: Attack, Decay, Sustain, and Release. Attack is how quickly the sound rises from silence to its greatest volume. Decay is the time required for the sound to drop back down to a constant level, called Sustain. The sound then falls off to silence, or Release. Registers 11 and 12 control how long each stage of the envelope lasts. The range of this number can be from 0-65535. The incoming clock frequency (2 Megahertz) is divided by 256 and then the result is again divided by the value contained in the two registers. Register 13 allows the programmer to select one of ten waveforms for the envelope (see the table on page vii).

Registers 14-15

These registers have nothing to do with sound and should not be used. The registers are used by the ST Computer as I/O ports.

Examples

1. SOUND [0 28 1 1 7 62 8 8 130 50 7 63 255 0]

   Result: A 440 Hz note A will be played for 1 second on voice A.

   Comments: Registers 0 and 1 are loaded with 28 and 1 respectively. From figures 1 and 3 we can determine that these commands put the tone period for a note A in octave 4 into the tone register for voice A. We then enable voice A by placing a 62 into register 7. The volume is then set to 8 with the fourth command. By using command 130 with 50 as an argument, we set the tone's duration to 1 second. Finally the sound is terminated by putting a 63 into register 7.

2. SOUND [0 28 1 1 2 194 3 1 4 123 5 1 7 56 8 8 9 8 10 8 130 50 7 63 255 0]

   Result: A chord of A, C#, and E will be played for 1 second.

   Comments: This example works the same way that the previous example does except that three voices are enabled instead of one.

3. SOUND [0 0 1 1 7 62 8 8 128 1 129 0 1 255 7 63 255 0]

Result: A "ramping" sound effect will be created.

Comments: As before, the first four commands enable voice A and set its volume. The next command, 128, places a 1 into a temporary register. Command 129 then follows taking 0, 1, and 255 for arguments. The first argument tells the sound chip to put the contents of the temporary register, in this case a 1, into register 0. The next two arguments tell the chip to increment this temporary register by 1 and to finish when it has reached a value of 255. The resulting sound ramps voice A from an A# to a C in octave 4.

Calculating Register Values

Two formulas can be used to calculate register values for a desired note.

Frequency = 440/(1.059463 ^ index)
Tone period = 2,000,000/(16 x frequency)

The values for the index are determined by how many notes apart the desired note is from a 440hz "A." For example, a note C in the same octave as the 440hz A would have an index of 9. A note B would have an index of −2.

Example

Calculate the register value of a note "A" which is one octave lower than the 440hz "A."

Index = 12
Frequency = 440/(1.059463 ^ 12) = 220
Tone period = 2,000,000/(16 x 220) = 568.18

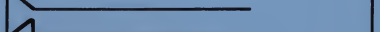
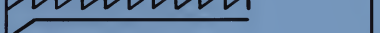
Now to calculate the value of the coarse register, take the integer part of this:

$$\frac{568}{256} = 2$$

The fine register value will contain this:

568 − (2 x 256) = 568 − 512 = 56

REGISTER $0D WAVEFORM CONTROL

	Control Bits				
	B3	B2	B1	B0	Selected Waveform Shape
DECIMAL	CONTINUE	ATTACK	ALTERNATE	HOLD	
0	0	0	X	X	
4	0	1	X	X	
8	1	0	0	0	
9	1	0	0	1	
10	1	0	1	0	
11	1	0	1	1	
12	1	1	0	0	
13	1	1	0	1	
14	1	1	1	0	
15	1	1	1	1	

0 - Off
1 - On
X - Not Used

Envelope Period
(duration of one cycle)

Frequency Values

The following table provides the frequency values for notes two octaves above and below middle C.

Note	Octave	Desired Frequency	Fine Tune	Coarse Tune
B	5	987.77	127	0
A#	5	932.33	134	0
A	5	880.00	142	0
G#	5	830.61	150	0
G	5	783.99	159	0
F#	5	739.99	169	0
F	5	698.46	179	0
E	5	659.26	190	0
D#	5	622.25	201	0
D	5	587.33	213	0
C#	5	554.37	225	0
C	5	523.25	239	0
B	4	493.88	253	1
A#	4	466:16	12	1
A	4	440.00	28	1
G#	4	415.30	45	1
G	4	392.00	63	1
F#	4	369.99	82	1
F	4	349.23	102	1
E	4	329.63	123	1
D#	4	311.13	146	1
D	4	293.66	170	1
C#	4	277.18	195	1
C	4	261.63	222	1
B	3	246.94	250	1
A#	3	233.08	24	2
A	3	220.00	56	2
G#	3	207.65	90	2
G	3	196.00	126	2
F#	3	185.00	164	2
F	3	174.62	204	2
E	3	164.81	246	2
D#	3	155.56	36	3
D	3	146.83	83	3
C#	3	138.59	134	3
C	3	130.81	188	3

Note	Octave	Desired Frequency	Fine Tune	Coarse Tune
B	2	123.47	244	3
A#	2	116.54	49	4
A	2	110.00	112	4
G#	2	103.83	180	4
G	2	98.00	9	5
F#	2	92.50	71	5
F	2	87.31	152	5
E	2	82.41	237	5
D#	2	77.78	71	6
D	2	73.42	167	6
C#	2	69.30	12	7
C	2	65.41	119	7

AY-3-8910 Registers

REGISTER		BIT B7	B6	B5	B4	B3	B2	B1	B0
0	CHANNEL A FREQUENCY	8 BIT FINE TUNE A							
1						COARSE TUNE A			
2	CHANNEL B FREQUENCY	8 BIT FINE TUNE B							
3						COARSE TUNE B			
4	CHANNEL C FREQUENCY	8 BIT FINE TUNE C							
5						COARSE TUNE C			
6	NOISE PERIOD			5 BIT PERIOD CONTROL					
7	VOICE ENABLE	$\overline{\text{IN}}$/OUT			$\overline{\text{NOISE}}$			$\overline{\text{ZONE}}$	
		IOB	IOA	C	B	A	C	B	A
8	CHANNEL A VOLUME				M	L3	L2	L1	LO
9	CHANNEL B VOLUME				M	L3	L2	L1	LO
10	CHANNEL C VOLUME				M	L3	L2	L1	LO
11	ENVELOPE PERIOD	8 BIT FINE TUNE E							
12		8 BIT COARSE TUNE E							
13	ENVELOPE SHAPE/CYCLE			CONTINUE		ATTACK		ALTERNATE	HOLD
14	I/O PORT A	8 BIT PARALLEL PORT A							
15	I/O PORT B	8 BIT PARALLEL PORT B							

Atari Corp., Sunnyvale, CA 94086

Printed in Taiwan
C026262 Rev. A

YES, enter my subscription to the ATARI EXPLORER:

☐ 6 issues . . . $15.00* ☐ 18 issues . . . $39.95* SAVE 25%

☐ Payment Enclosed ☐ Visa ☐ MasterCard

Name _____

Address _____

City _____

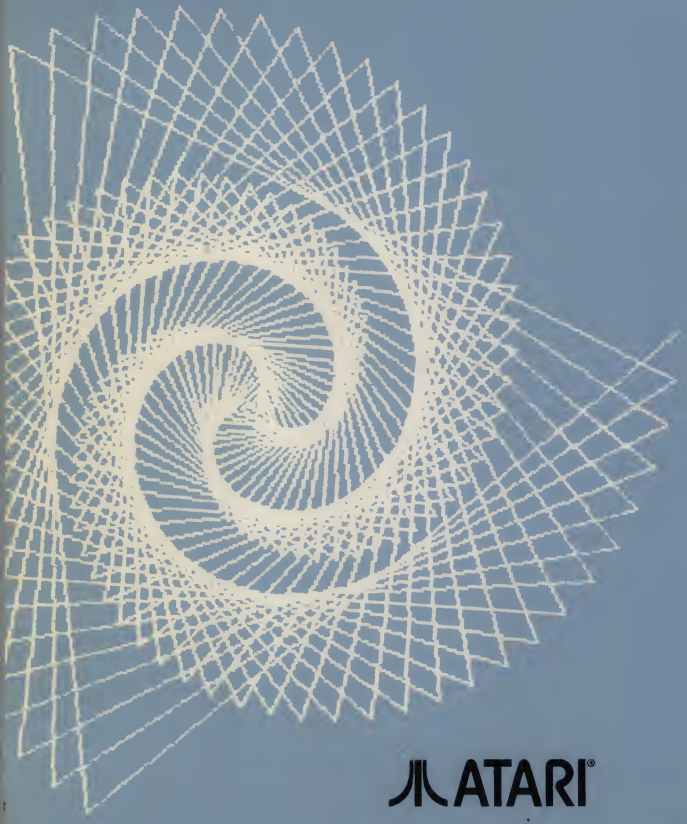
State _____ Zip _____

Acct No. _____

Expiration Date _____

Signature _____

*This rate limited to USA and its Possessions. Canadian subscriptions, add $5 for postage and handling. Overseas subscriptions, add $10.00/6 issues; $30.00/18 issues. Payment due in US funds.



ATARI®

Atari Corp., Sunnyvale, CA 94086

Printed in Taiwan
CO26106-001/D
Rev. C K. I. 2. 1986